MODELAGEM CONCEITUAL

e Projeto de Bancos de Dados

Como observar, abstrair e modelar dados e objetos

...ado em um programa de capacitação profissional aprimorado ao longo de dez anos, este livro possui uma abordagem didática e de fácil compreensão por qualquer pessoa com conhecimento mínimo de informática.

Seu enfoque não requer conhecimento prévio de modelagem e projeto de bancos de dados, nem conhecimento de metodologia ou ferramenta CASE.

Modelagem Conceitual e Projeto de Bancos de Dados ensina a estudantes, analistas de sistemas e administradores de bancos de dados a projetar e manipular estruturas de bancos de dados para uso comercial e pessoal.

Aborda detalhadamente o tema modelagem conceitual, procurando desenvolver habilidades de análise, abstração e síntese dos elementos observados durante o processo de modelagem, seja no ambiente convencional de modelagem de dados sugerida pelas metodologias de análise estruturada, ou no ambiente orientado a objetos.

Propõe a obtenção do modelo lógico de dados (projeto do BD) a partir da derivação do modelo conceitual e não através da modelagem direta

Apresenta outras aplicabilidades do processo de modelagem conceitual que não de projeto de BD, expandindo o universo dos usuários dessa técnica.

Aborda diversas metodologias de modelagem, comparando-as e nivelando-as, possibilitando do leitor a escolha adequada

Utiliza uma abordagem diferenciada que tem sido avaliada de extrema eficácia em projetos de consultoria e em programas de treinamento realizados em grandes empresas.

Paulo Cougo

É sócio-diretor da Tree Tools Informática, empresa sediada em Curitiba. Pós-graduado em Análise de Sistemas aplicada à Administração de Empresas pela PUC/PR, é consultor e instrutor de Administração de Dados e Tecnologia de Bancos de Dados, tendo atuado durante longo período como instrutor dos cursos regulares de Modelagem de Dados da TELEPAR. Sua experiência profissional, enriquecida pelos anos de ensino em faculdades de Curitiba, vem consagrando seu programa de treinamento e consultoria junto a grandes empresas.

ISBN 85-352-0156-9

9 788535 201561

Nível

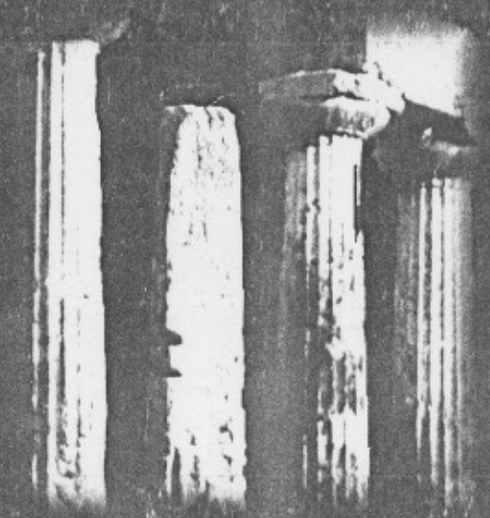
Iniciante — Intermediário — Avançado

Categoria: Análise de Sistemas

http://www.campus.com.br

Cougo — Modelagem Conceitual e Projeto de Bancos de Dados

Paulo Cougo



MODELAGEM CONCEITUAL

e Projeto de Bancos de Dados

EDITORA CAMPUS

© 1997, Editora Campus Ltda.

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/73.
Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.
Todo o esforço foi feito para fornecer a mais completa e adequada informação. Contudo a editora e o(s) autor(es) não assumem responsabilidade pelos resultados e uso da informação fornecida. Recomendamos aos leitores testar a informação antes de sua efetiva utilização.

Capa:
Wladimir Melo

Copyleaque
Raul Cardoso de Araujo

Editoração Eletrônica:
Rio Texto

Revisão Gráfica:
Isabel Cristina Rodrigues
Angela Castello Branco

Projeto Gráfico:
Editora Campus Ltda.
A Qualidade da Informática.
Rua Sete de Setembro, 111 – 16º andar
20050-002 Rio de Janeiro RJ Brasil
Telefone: (021)509-5340 FAX (021)507-1991
E-Mail: info@campus.com.br

ISBN 85-352-0158-0

Ficha Catalográfica
CIP-Brasil. Catalogação-na-fonte.
Sindicato Nacional dos Editores de Livros, RJ

Cougo, Paulo Sérgio

C892m Modelagem conceitual e projeto de bancos de dados / Paulo Cougo. — Rio de Janeiro: Campus, 1997.

ISBN 85-352-0158-0

1. Projeto de sistemas. 2. Programação orientada a objetos (Computação). 3. Banco de dados – Desenvolvimento. I. Título

97-0226

CDD — 005.1
CDU — 681.3

97 98 99 00 5 4 3 2 1 0

EDITORA AFILIADA

Sumário

INTRODUÇÃO 1
A Modelagem de Dados e a Orientação a Objetos 1
O Enquadramento de Objetos em Classes e Entidades 3
Modelagem de Dados e Projeto de Banco de Dados 4
Repensando a Modelagem 5
Referências Bibliográficas 5

Capítulo 1
CONCEITOS BÁSICOS 7
Definição de Modelo 7
O Papel do Objeto Observado 8
A Modelagem como um Processo Estruturado 10
O Processo de Modelagem 12
Especificação dos Requisitos 12
Execução da Modelagem dos Dados 14
Objetivos do Modelo de Dados 18
Representar um Ambiente Observado 19
Servir de Instrumento para Comunicação 21
Favorecer o Processo de Verificação e Validação 22
Capturar Aspectos de Relacionamento entre os Objetos Observados 23
Servir como Referencial para a Geração de Estruturas de Dados . . 23

Capítulo 2
NÍVEIS DE MODELAGEM 25
Definição de Níveis 25
Modelo Conceitual de Dados (MCD) 28
Modelo Lógico de Dados (MLD) 29
Modelo Físico de Dados (MFD) 30
Integração da Arquitetura de Três Níveis com a Abordagem E-R . . . 31

Capítulo 3
MODELO E-R 33
A Abordagem Entidade-Relacionamento 33
O Embasamento para o Modelo E-R 33
Reconhecimento dos Elementos no Ambiente Observado 34

Entidades 37
Identificação de Objetos (Entidades ou Classes) 37
A Dicionarização dos Objetos Modelados 46
Considerações sobre Entidades Fortes e Entidades Fracas 53
Atributos 57
O Papel dos Atributos 57
Classificação dos Atributos Quanto a Sua Finalidade 60
Atributos Nominativos 61
Dicionarização dos Atributos 64
Relacionamentos 66
Identificação dos Relacionamentos 66
Caracterização dos Relacionamentos 75
Dicionarização de Relacionamentos 106
Elementos de Caracterização Semântica Adicionais 108
Estrutura de Generalização e Especialização 108
Estrutura de Agregação 131
Notações Gráficas Utilizadas na Modelagem Conceitual 138
Notação IDEF1X 138
Notação de Bachman 143
Notação de Setas 145
Notação "Pé-de-Galinha" (*Crow's foot*) 146
Notação de Martin (Engenharia da Informação) 146
Notação de Peter Chen 151
Qual a Melhor Notação? 155
Referências Bibliográficas 156

Capítulo 4
DERIVAÇÃO DO MODELO LÓGICO 157
Passos para a Derivação do Modelo Lógico 157
Adaptações no Modelo Lógico Derivado 158
Conceitos Utilizados na Derivação de Modelos Relacionais 159
Implementação do Modelo Relacional 161
Conceitos 162
Regras de Derivação 171
Normalização 173
Benefícios da Normalização 176
Derivação de Relacionamentos 220
Derivação de Auto-relacionamentos 231
Derivação de Estruturas de Generalização-Especialização 237
Derivação de Agregações 247

Capítulo 5
RESTRIÇÕES NO MODELO LÓGICO 255
Restrições de Domínio 255
Restrições de Integridade 258
Restrições de Implementação 262

Capítulo 6
O AMBIENTE CORPORATIVO 265
Obtendo o Modelo Corporativo 265
A Abordagem Top-down 266
A Abordagem Bottom-up 268
Modelo de Dados de Aplicação 268
Modelo de Dados Setorial 269
Modelo de Dados Corporativo 270
Compatibilização dos Modelos de Dados 270
Nivelamento dos Modelos de Dados 272
A Abordagem Top-down x Bottom-up 274

E FINALMENTE 277

BIBLIOGRAFIA 279

ÍNDICE 281

Apresentação

A modelagem de dados tem sido tratada, por grande parte dos autores, sob o ponto de vista de implementação das estruturas de dados observadas. Isso tem levado àqueles que se iniciam nessa atividade a ter uma visão distorcida do processo de modelagem e, por conseguinte, a não obterem os resultados potenciais da aplicação dessa técnica. Boa parte da literatura disponível no mercado, por influência do perfil de seus autores, tem dado grande ênfase ao processo de implementação lógica (relacional quase sempre) e sacrificado o que chamamos de *base para a modelagem*, que é o processo conceitual.

A experiência tem nos demonstrado que grande parte da dificuldade encontrada por aqueles que se iniciam na atividade de modelagem de dados é resultante da excessiva ênfase dada ao aspectos tecnológicos, o que faz com que não consigam, efetivamente, "enxergar o mundo que modelam à sua frente". Essas pessoas, presas a preocupações tais como normalização, redundância, chaves de acesso e tantas outras exigências tecnológicas, se tornam incapazes de abstrair os conceitos do mundo que estão observando e modelando.

Durante os mais de dez anos de atuação na área de modelagem de dados, pudemos constatar que o único meio eficaz para reverter essa situação é restabelecer a devida importância da modelagem conceitual, fazendo-a tornar-se o ponto de partida para o processo de modelagem lógica de dados.

Assim, foi elaborado este livro onde é abordado de maneira abrangente o tema Modelagem Conceitual. Este texto é baseado em um conteúdo programático desenvolvido e aplicado, já há vários anos, visando à capacitação de profissionais para a área de modelagem de dados e administração de dados.

Entretanto, durante os trabalhos, percebemos que muito do que estava sendo tratado direcionado somente à modelagem de dados poderia ser, também, aplicável à modelagem de informações (enfoque dado pela orientação a objetos). Por isso, você irá perceber que este livro não visa somente à abordagem convencional de modelagem de dados sugerida pelas metodologias de análise estruturada, análise essencial ou engenharia da informação, mas também atende à abordagem orientada a objetos.

O enfoque que daremos à modelagem conceitual de dados aproximará bastante as técnicas utilizadas e os resultados obtidos no ambiente convencional e no ambiente de orientação a objetos. Procuraremos apresentar meios para que os modelos de classes e os modelos de dados incorporem características semânticas comuns. Isso fará com que as técnicas aqui apresentadas para a derivação de modelos lógicos de dados também sejam aplicáveis aos modelos de classes. Como hoje, praticamente, todos os projetos orientados a objetos acabam por implementar seus modelos de classes através de modelos de dados relacionais, deverão ser conhecidas as técnicas e meios para que essa derivação não seja realizada de modo intuitivo ou para que não sejam necessárias novas concepções de modelos durante a fase de projeto.

Assim, esteja você envolvido com o ambiente convencional de desenvolvimento de sistemas ou com o ambiente orientado a objetos, esperamos poder contribuir de algum modo com os tópicos que aqui serão abordados.

Paulo Sérgio Cougo
1º de março de 1997

Introdução

A Modelagem de Dados e a Orientação a Objetos

A modelagem de dados, originalmente centrada somente sob o enfoque da abordagem *entidade-relacionamento* proposto por Peter P. Chen na década de 1970, tem evoluído, com o passar dos anos, para enfoques que a tornam muito mais próxima dos conceitos e do ambiente de orientação a objetos. Durante essa evolução, têm sido agregados novos elementos ao modelo E-R, que tornam a técnica mais rica em semântica e, portanto, aplicável a novas finalidades.

A modelagem de informações, por exemplo, apresentada no livro *Análise de Sistemas Orientada para Objetos*[1] por Shlaer & Mellor já demonstra o novo papel da modelagem de dados aplicada ao processo de mapeamento das características estáticas dos objetos. Os autores abordam extensões do modelo entidade-relacionamento, voltando-o mais para o aspecto semântico, transformando-o em meio de identificação e mapeamento de objetos e classes.

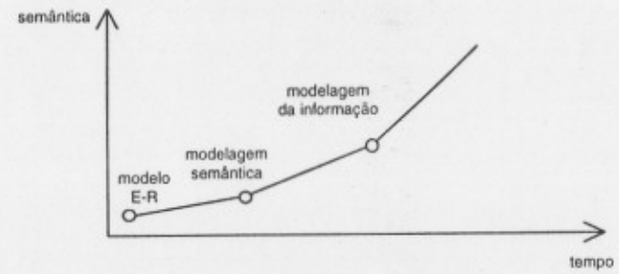
É bem verdade que, neste caso, estaremos enfocando somente uma parte do todo que, realmente, pretendemos atingir quando falamos em mapeamento dos objetos. Estaremos, ao aplicar a modelagem de informações, desconsiderando os aspectos dinâmicos ligados aos objetos mapeados. Por outro lado, quanto ao mapeamento das características estáticas teremos grande parte, senão a totalidade, de nossas necessidades atendidas.

Coad & Yourdon também ressaltam o papel da modelagem de informações junto à abordagem orientada a objetos afirmando literalmente em seu livro *Análise Baseada em Objetos*:[2] "A análise orientada a objetos é baseada nos melhores conceitos de Modelagem de Informações..." Apresentam também, nesse mesmo livro, o quadro a seguir, no qual pode-se melhor visualizar o enquadramento de uma e outra abordagem quanto a seu atendimento aos princípios da administração da complexidade, princípio este que norteia todo o enfoque de orientação a objetos.

Princípios	Modelagem de Informações	Orientação a Objetos
abstração de procedimentos		X
abstração de dados		X
encapsulamento		X
herança		X
associação	X	X
comunicação de mensagens		X
método de organização: objetos e atributos	X	X
método de organização: todo e partes	X	X
método de organização: classes, membros e distinção deles	X	X
escala		X
categoria de comportamento: causa imediata		X
categoria do comportamento: mudança com o tempo		X
categoria de comportamento: similaridade de funções		X

James Rumbaugh, em seu livro *Modelagem e Projetos Baseados em Objetos*,[3] compara diversas metodologias de vários autores à sua abordagem, a OMT. Ao posicionar a OMT em relação à modelagem da informação afirma: "A modelagem de objetos através da técnica OMT também é uma forma melhorada da modelagem E-R."

Esses exemplos mostram que, dentro do enfoque de orientação a objetos, existe uma lacuna que necessariamente deve ser preenchida por algum tipo de abordagem que nos leve ao mapeamento das estruturas estáticas dos objetos (os objetos e seus atributos, estruturas todo e partes, classes e membros). A modelagem de dados, modelagem semântica ou modelagem de informações, dependendo do nível evolucionário da técnica aplicada, atende a essa finalidade integralmente e para tal deve ser aplicada.



O Enquadramento de Objetos em Classes e Entidades

Usaremos durante o transcorrer das explanações o termo *objeto* para denominar *elementos observados*, sem nos restringirmos ao formalismo sob o ponto de vista de enquadramento de "classes", mas sendo totalmente compatível com esse paradigma. Estaremos mais freqüentemente voltados à abordagem entidade-relacionamento, na qual a reunião desses objetos dará origem ao conceito de "entidade" e não ao conceito de "classe" como na orientação a objetos. Entretanto, se esta última for sua visão de maior interesse, tenha total liberdade de assim entender e considerar. Você verá que tanto em um como em outro caso haverá compatibilidade entre os conceitos desenvolvidos.

Veremos claramente que as abstrações feitas em um e outro ambiente nos levam por um mesmo caminho durante o processo de modelagem. Quanto mais próximos estivermos de um modelo conceitual, mais afinidades e similaridades encontraremos no processo de modelagem de dados (modelo entidade-relacionamento) ou de informações (modelo de objetos). Esse processo, em ambos os casos, estará baseado na execução dos seguintes passos:

1. Observar os elementos de um ambiente.
2. Elaborar conceitos sobre esses elementos.
3. Caracterizá-los.
4. Abstrair características.
5. Reuni-los em conjuntos que os denominem.
6. Representá-los.
7. Defini-los.
8. Manipulá-los.

Isso, desde já, realça a importância de se *buscar a construção de um modelo conceitual antes de se partir para a elaboração de um modelo lógico*. Se para você isso ainda não está muito claro, prossiga assim mesmo, pois

durante o desenvolvimento do assunto atingiremos o nível de conceitu[illegible]o necessário à sua compreensão.

Apresentamos, no quadro a seguir, os termos que estarão sendo utilizados na abordagem dada pelo ambiente convencional e pelo ambiente orientado a objetos. Esses termos, em alguns instantes, serão claramente diferenciados e, em outros, quando a similaridade os faça serem substituíveis, terão referências indistintas. Estaremos conceituando cada um desses termos nos capítulos em que isso seja apropriado. Neste instante, basta que se tenha claro a que abordagem eles pertencem, a que se referem e qual a equivalência entre eles.

Conceitos Utilizados na Abordagem Convencional e na Orientada a Objetos

Termos	Modelo de Dados (convencional)	Modelo de Informação (orientado a objetos)
instâncias	ELEMENTOS INDIVIDUALIZADOS OU OCORRÊNCIAS	OBJETOS
conjunto formado pela agregação de instâncias semelhantes	ENTIDADE	CLASSE
características das instâncias dos conjuntos	ATRIBUTO	ATRIBUTO
envolvimento entre as instâncias dos conjuntos	RELACIONAMENTO	ASSOCIAÇÃO

Modelagem de Dados e Projeto de Banco de Dados

A modelagem de dados tem sido um assunto tratado, na maioria das vezes, associado à atividade de projeto de bancos de dados. Grande parte das referências bibliográficas demonstra isso, basta você consultá-las. Entretanto, para que se possa atingir realmente o nível de compreensão e assimilação necessário, devemos procurar desvincular essas duas áreas. Devemos entender que:

> *A modelagem de dados* **não é** *uma ferramenta* **só** *para o projeto de bancos de dados!*

Se você está interessado em uma ferramenta que o auxilie no processo de projeto de banco de dados, com certeza as técnicas de modelagem de dados que serão vistas neste livro serão de grande valia.

Veremos no deco[illegible]r do assunto, entretanto, que a modelagem de dados se aplica a outras finalidades também. Pode ser uma ferramenta extremamente útil para a área de O&M, para processos de reengenharia, para definição de regras de negócio, etc.

Mas, antes que avancemos mais, independentemente de suas intenções, precisamos definir alguns pontos importantes. Pontos que, muitas vezes, até os mais convictos especialistas têm parado para se perguntar, tais como:

- O que significa modelar?
- É preciso mesmo modelar?
- Como devemos modelar?
- Por que devemos modelar?
- Para que serve o modelo após gerado?
- O que deve ser representado no modelo?

Repensando a Modelagem

Você já parou alguma vez para se perguntar se, realmente, a modelagem de dados é necessária ou se este é só mais um formalismo que acaba não nos levando muito longe? Será que, realmente, o procedimento mais produtivo não seria partir para o micro e usar telas amigáveis, assistentes (*wizards*), modelos (*templates*) e tantos outros recursos, criar logo de uma vez todas as tabelas que nos parecem necessárias e sair então utilizando-as? Já pensou?

O objetivo dos próximos capítulos será capacitá-lo a responder a essas e outras questões comprovando, finalmente, que *modelar é preciso...*

Referências Bibliográficas

[1]SHLAER & MELLOR. *Análise de Sistemas Orientada para Objetos*. São Paulo: MacGraw-Hill, 1990.

[2]COAD & YOURDON. *Análise Baseada em Objetos*. Rio de Janeiro: Campus, 1992.

[3]RUMBAUGH, James. *Modelagem e Projetos Baseados em Objetos*. Rio de Janeiro: Campus, 1994.

1

Conceitos Básicos

Definição de Modelo

Para que possamos falar em modelagem de dados, no sentido mais amplo do termo, temos de ter claro o que representa cada um dos elementos envolvidos nesse processo. Assim, vamos inicialmente explorar a fundo o conceito de *modelo*. Parece trivial, a todos nós, o que é um modelo, mas, por hora, vamos esquecer o aspecto puramente informático ao qual estamos presos e explorar outras possibilidades interessantes.

A seguir, temos uma definição, entre as várias possíveis, que particularmente parece bastante aplicável a grande parte dos casos que iremos enquadrar.

> **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 partes.*

Para validar essa definição, vamos partir de um caso bastante conhecido de todos nós. Consulte seu jornal de domingo, qualquer que seja. Olhe para um anúncio que tenha uma foto da maquete de um apartamento, ou para o desenho de uma planta baixa de um imóvel qualquer no caderno imobiliário. Você consegue perceber modelos à sua frente? O que nos diz a definição acima?

"Um modelo é uma representação abstrata e simplificada."

Sim, inclusive algumas vezes essas maquetes ou plantas são tão abstratas e simplificadas que nem sequer chegamos a ter certeza do imóvel que estamos comprando, mas de qualquer modo você pode identificar e escolher entre uma quitinete ou um apartamento de quatro quartos com varanda mesmo antes de ter ido até o local da construção, certo? Isso é um modelo!

"...que pode explicar ou testar o comportamento de um sistema real..."

Sim, mais uma vez você não precisa ter o imóvel construído à sua frente para perceber que um quarto, sem janelas e cuja porta dá de frente para a cozinha não vai satisfazer suas necessidades pessoais de ventilação. Isso é um modelo!

Logo, o modelo não é o objeto real mas algo que o representa, com maior ou menor fidelidade. Faz com que pela sua observação e manipulação tenhamos nossas necessidades de conhecimento e conceituação sobre um objeto satisfeitas.

Encontramos outros exemplos fora da informática? Inúmeros no nosso dia-a-dia. Tantos que nem nos apercebemos. Mas utilizamo-nos deles e sem eles teríamos algumas dificuldades em perceber certos conceitos.

- O *manequim* na vitrine.
- A *foto* de um conjunto estofado.
- Um *aeromodelo* sendo usando em testes de aerodinâmica.
- O *desenho* em perspectiva de sua nova cozinha.
- O *molde* de uma nova roupa obtido em uma revista.
- O *memorial descritivo* de seu novo apartamento.

Assim como esses, vários outros exemplos poderiam ser dados para ilustrar o uso de modelos em nosso cotidiano. O importante, contudo, é perceber que através de algum meio, seja uma *maquete*, um *desenho*, uma *fotografia*, uma *descrição*, etc. podemos antecipar, ou substituir a existência de uma realidade qualquer. Através de um elemento semelhante, em escala ou não, representamos um objeto desejado e assim podemos percebê-lo e entendê-lo.

O Papel do Objeto Observado

Em todos os exemplos que demos anteriormente, sempre estivemos apresentando modelos de algum *objeto* que, em algum instante, serviu como referência para sua criação (a maquete teve o apartamento, o manequim baseou-se em um ser humano). A esse elemento de referência chamaremos *objeto observado*.

O termo "objeto" é usado, de modo genérico, para caracterizar qualquer coisa, pessoa, ambiente, conceito, etc. Mesmo que estejamos trabalhando em um ambiente convencional de desenvolvimento de sistemas, não estaremos errados se imaginarmos que estamos tratando e modelando objetos. Descobre-se, então, aqui, o primeiro forte vínculo entre a já tradicional orientação a dados e a nova orientação a objetos.

O objeto observado, sem sombra de dúvidas, é o ponto de partida para qualquer processo de modelagem, de dados ou não. Temos de ter um objeto a reproduzir, seja ele concreto ou imaginário.

Assim, o arquiteto, ao desenhar a planta baixa de um novo apartamento, está reproduzindo um objeto imaginário que, pelo menos, em sua imaginação já existe. Existe em forma de lembranças de outros imóveis já vistos ou projetados, em forma de conceitos assimilados na faculdade ou em livros, em forma de idéias, ou outras referências quaisquer. Por outro lado, também poderíamos estar falando de um desenho de uma planta baixa feito após uma vistoria ao local onde foi construído um prédio. Nesse caso, estaríamos modelando um objeto concreto e não mais um imaginário.

Esses conceitos parecem triviais mas, do ponto de vista de modelagem de dados, são extremamente importantes. O ponto principal é que a modelagem de dados necessita também, obrigatoriamente, de um objeto a ser observado, seja ele real ou imaginário.

Você, ao ter contato com o mundo da modelagem de dados, já deve ter ouvido, ou ainda ouvirá, por mais de uma vez, a afirmação: "...temos que modelar o mundo real...". Após o que já apresentamos poderíamos redefinir essa expressão como:

> *Temos que modelar o* **mundo observado**, *seja ele real ou imaginário.*

Assim, se alguém tiver criado algum modelo de dados, certamente terá reproduzido, como o arquiteto, algo que observou, concreto ou imaginário. Por isso, e só por isso, o processo de levantamento de dados, investigação e análise dos dados é tão importante para a modelagem de dados. Se não captarmos informações que nos caracterizem devidamente os objetos, não conseguiremos "observá-los" e logo nosso modelo não os contemplará.

Mas, ao observarmos os objetos de um mundo real ou imaginário:

- O que perceberemos?
- Como eles nos serão apresentados?
- O que deveremos buscar?

Veremos mais a frente que uma das atividades mais complexas no processo de modelagem de dados é a compatibilização entre as características observadas e as reproduzidas (ou modeladas). Isso se dá porque os objetos observados têm *características próprias* que, sejam qualidades ou defeitos, acabarão por serem percebidas por nós. E mais, os objetos observados muitas vezes têm um comportamento dinâmico e são multifacetados. Parece complicado, não? Para esclarecer, vamos novamente buscar exemplos fora da informática e, depois, contrapô-los com os da informática.

A Modelagem como um Processo Estruturado

Imaginemos um processo de modelagem, como por exemplo o desenho ou pintura de um quadro. Provavelmente, você nunca tenha pensado na pintura como um processo de modelagem, mas se observá-la verá que é. Depois de conhecer melhor o processo de aprendizagem de desenho e pintura de "modelos vivos" (pessoas), pode-se perceber que os alunos, ao reproduzirem o objeto que observam, têm que lidar com certas peculiaridades:

1. Cada pessoa que posa para um grupo de alunos tem suas características próprias como altura, peso, estatura, feição, contornos, apesar de todas seguirem basicamente um mesmo padrão.
2. Algumas das características dessas pessoas embelezariam o trabalho final se fossem reproduzidas, e outras tais como cicatrizes, manchas na pele, etc., que existem no modelo observado, podem ou não ser reproduzidas na tela a ser pintada, com a possibilidade de, em alguns casos, prejudicar a estética do produto final.
3. As pessoas que posam, devido ao grande período de tempo necessário para que todas as suas características sejam reproduzidas, alteram levemente sua posição durante pequenos intervalos de descanso. Isso faz com que os alunos tenham que se adaptar a essas mudanças alterando seus quadros ou desconsiderando pequenos detalhes que tenham se modificado.
4. Cada aluno tem um ponto de vista diferente ao realizar sua reprodução, pois estão posicionados em pontos diferentes da sala, alguns de frente, outros de lado e outros atrás do modelo, fazendo com que alguns percebam detalhes que outros estão impossibilitados de perceber num dado instante, tornando o resultado final, apesar de coerente, particularmente diferente para cada um deles.

Sim, mas o que tem a modelagem de dados a ver com tudo isso? Muito, pois podemos aproveitar tanto os problemas como as soluções por eles adotados para tratar de dificuldades que enfrentamos durante o processo de modelagem dos dados. Assim, vejamos as similaridades:

1. Cada ambiente com o qual nos depararmos para modelar terá suas próprias características. Por mais que acreditemos que já as conhecemos de outras experiências anteriores, devemos estar preparados para as particularidades que certamente existirão e que tornarão cada trabalho e cada modelo gerado semelhante em sua essência, mas único em seus detalhes.

2. O fato de observarmos qualidades e defeitos, ou conformidades e anomalias, no ambiente que pretendemos modelar, é algo que devemos estar preparados para tratar de algum modo. Iremos preservar somente as qualidades corrigindo os defeitos ou nosso modelo deverá ser fiel e também retratar essas anomalias?
3. O trabalho de modelagem por ser, em alguns casos, extenso e demorado poderá se defrontar com a situação em que, ao chegarmos a certa altura do modelo, a porção que anteriormente havia sido modelada já sofreu impactos por pequenas mudanças introduzidas no ambiente observado.
4. A visão que poderá ser obtida dos objetos ou do ambiente que desejamos modelar pode ser fortemente dependente de pontos de vista diferentes que nos sejam apresentados, ou mesmo, do ponto de vista que estejamos percebendo. Isso fará com que vários modelos, cada um com diferentes características, possam existir e ser coerentes entre si, por retratarem visões diferentes, todavia deverão preservar todos a mesma essência.

Como podemos perceber, há pontos em comum nos dois casos abordados e, existindo a similaridade, ficam sendo também válidas as considerações, estratégias e soluções adotadas pelos artistas em suas pinturas:

1. Podemos partir de padrões, ou preconceitos com os quais aceleraremos a produção do nosso resultado final. Assim como um aluno sabe as proporções e estruturas básicas de um corpo humano e as utiliza para ajustar seu desenho a cada novo quadro, também nós poderemos aproveitar as experiências de outros modelos já realizados sobre o mesmo objeto ou ambiente para, a partir dos conceitos já conhecidos, acelerar e favorecer a criação do novo modelo, respeitando e observando sempre as novas particularidades observadas.
2. Devemos procurar nos posicionar, com a orientação do interessado no modelo a ser gerado, sobre qual deve ser nossa estratégia com relação à reprodução das qualidades e defeitos. Mantê-los? Corrigi-los? Suprimi-los?
3. A cada caso, devemos verificar se as mudanças no objeto ou ambiente observado são significativas para a produção de nosso resultado final e se a fase em que nos encontramos viabiliza uma grande mudança quanto ao conteúdo do que já temos modelado. Será uma questão a ser estudada e que poderá, em alguns casos, tanto nos levar ao reinício de todo o trabalho como, em outros casos, a desprezar as mudanças percebidas.

4. Para que se tenha uma visão completa de todos os detalhes poderão ser adotadas duas estratégias. A primeira será procurar contar durante os trabalhos de modelagem com pontos de vista diferentes tentando, tanto quanto possível, agrupá-los para a formação de um único produto. A segunda será contar com mais de uma representação, em diferentes níveis, para poder transmitir todos os conceitos captados. Esse último recurso será melhor explorado quando estivermos falando de modelos conceituais e modelos lógicos.

O Processo de Modelagem

A obtenção de um modelo a partir de um conjunto de objetos observados deve levar também em conta alguns quesitos. Cada um deles contribuirá decisivamente para que o produto gerado pela modelagem seja o esperado. Em ordem de atendimento, os quesitos a serem satisfeitos são:

Especificação dos Requisitos

Alguns pontos importantes devem ser definidos mesmo antes de se iniciarem os trabalhos de modelagem, para que assim sejamos orientados na geração do produto esperado. Dentre esses pontos podemos enumerar:

- **Abrangência.** Definir a abrangência, ou escopo, dos trabalhos é de vital importância. Precisamos desse referencial para poder definir que objetos teremos que buscar em nosso processo de observação. Um artista precisa saber se deve pintar toda uma paisagem ou só a árvore que está no meio do bosque à sua frente. O nível de trabalho, e de observação, será bastante diferente, assim como o produto a ser gerado. Expectativas irreais, ou maldefinidas, podem levar o melhor dos modelos a ser totalmente descartado por não retratar o que era desejado.

- **Nível de Detalhamento.** Mesmo que já tenhamos um escopo bem definido, precisamos saber, adicionalmente, quais são as expectativas com relação ao nível de detalhamento esperado para o produto final. Como no exemplo anterior, podemos saber que o que se deseja é a reprodução de todo o bosque, mas em que nível de detalhe? Somente desejamos as linhas gerais? Os elementos predominantes? Tudo? Com a representação dos mínimos detalhes? Imagine, agora, esse panorama voltado para a área de modelagem de dados. Falaremos mais sobre esse assunto quando estivermos tratando de modelos conceituais e modelos semânticos. Aguarde.

- **Tempo para a produção do Modelo.** Podemos, também, já ter definidos o nível de detalhamento e a abrangência mas, mesmo assim, ainda teremos sérias dificuldades em produzir o modelo desejado se não conhecermos as expectativas de prazo para a conclusão do trabalho. Você, por certo, já ouviu falar na Capela Sistina, com inúmeras reproduções pintadas em seu teto por Michelangelo. Um bom exemplo de um grande trabalho de modelagem. Mas em que condições e prazo foi executada? E se a necessidade de conclusão da obra fosse de um mês entre o início e o final do trabalho? Certamente o produto gerado não atenderia às expectativas, ou porque ficaria inacabado ou porque pecaria em qualidade. Assim também muitos projetos de modelagem de dados têm resultado em produtos que não atendem às expectativas por falha nesse aspecto.

- **Recursos Disponíveis.** Paralelamente ao tempo para a produção do modelo, deverá ser considerado um outro fator determinante a ser especificado desde o início dos trabalhos: a equipe a ser alocada para participar da modelagem. Quando falamos de equipe, estamos nos referindo tanto aos especialistas em modelagem como aos indivíduos que estarão fornecendo subsídios para tal. Isso é importante pois, naturalmente, respeitadas as limitações, quanto maior o número de participantes menor a expectativa de tempo para conclusão dos trabalhos. Voltando à Capela Sistina, apesar do trabalho imenso a ser realizado, provavelmente, que me desculpe Michelangelo, se um grupo de artistas, igualmente qualificados, tivesse se dedicado ao trabalho simultaneamente, teríamos todas as reproduções concluídas em um mês de trabalho.

Execução da Modelagem dos Dados

Supondo que já tenhamos tido todas as especificações de requisitos satisfeitas, não nos restará muito mais a fazer senão iniciar o trabalho de modelagem propriamente dito. A partir deste instante, entraremos em um processo cíclico de atividades que também terá seus quesitos a serem atendidos e que envolverá os seguintes passos:

1. A observação dos objetos.
2. O entendimento dos conceitos.
3. A representação dos objetos.
4. A verificação de fidelidade e coerência.
5. A validação do modelo.

Cada um desses passos fornece elementos para que o próximo passo possa ser iniciado, ou então mostra que existem anomalias ou falhas em sua execução, o que exigirá a volta aos passos anteriores. Esse ciclo será repetido

tantas vezes quantas forem as extensões agregadas ao modelo, seja em sua fase de construção ou em suas manutenções evolutivas, adaptativas ou corretivas.

Passo 1: A observação dos objetos

Sabemos que a definição da abrangência e do nível de detalhes será importante para o sucesso de nosso processo de modelagem. Entretanto, isso somente servirá como um referencial para que limitemos nossa busca, ou nosso foco de atenção.

Não imagine você que, uma vez definida a abrangência, todos os objetos que lhe serão apresentados pelos processos de levantamento de dados pertencerão ao escopo predefinido. Você, conforme sua habilidade em levantar dados, poderá escolher caminhos que o levem para mais perto do resultado esperado, mas, certamente, muito "ruído", ou "lixo" virá junto à massa de dados obtida por qualquer que seja o processo.

Saber "separar o joio do trigo" será uma atribuição sua, que com maior ou menor facilidade será executada caso a caso. A experiência, sem dúvida, poderá contribuir nesse aspecto, mas mesmo alguns *experts*, muitas vezes, se vêem confusos sobre se determinado objeto pertence ao modelo ou não. Aqui, mais uma vez, lembramos: diferentes pontos de vista poderão ser úteis para a identificação real dos objetos. Isso implica tanto a consulta de várias fontes como a avaliação de várias propostas de modelagem.

Mas, e para conhecer os objetos que deverão passar pelo processo de análise de dados, qual é a receita?

As técnicas de levantamento de dados podem contribuir bastante, sendo, basicamente elas, o nosso ponto de partida. Assim, através de entrevistas, reuniões, questionários, análise de documentos, análise de dados já estruturados em outros processos, etc., poderemos encontrar a quase totalidade de nossos objetos.

Também outro elemento pode ser acrescentado como fonte para identificação dos objetos: o conhecimento prévio, ou a experiência prévia. Como vimos anteriormente, padrões e preconceitos podem ser um bom referencial para percebermos a falta de "objetos típicos" de um ambiente que não nos tenham sido expostos pelo levantamento tradicional de dados.

Passo 2: O entendimento dos conceitos

A atividade de modelagem não é a representação dos objetos como normalmente possa parecer. Um conceito errado, que muitas vezes fazemos, é o de que quando vemos alguém desenhando um modelo de dados pensamos estar, naquele instante, se processando a modelagem dos dados. Na verdade nesse instante o que está ocorrendo é, única e exclusivamente, a representação dos objetos que já foram identificados, conceituados, entendidos e assimilados.

Por isso, o núcleo do processo de modelagem, ou o ponto onde realmente se processa a transposição do objeto observado para o objeto reproduzido, é o instante em que, ao analisar um objeto, conseguimos:

- Identificá-lo.
- Conceituá-lo.
- Entendê-lo.
- Assimilá-lo.

Essa seqüência de fases faz com que algo que até certo instante era desconhecido para nós passe a fazer parte de nosso conhecimento e seja incorporado ao conjunto de objetos de nosso domínio. Ao passar por essas fases, estaremos obtendo os elementos que definirão esse novo objeto, ou, mesmo que já seja um objeto conhecido, obtendo os elementos que definirão suas novas características. E esse é o processo para o qual um bom modelador de dados deve atentar. Desse modo se processa o crescimento profissional na área de modelagem de dados.

Passo 3: Representação dos objetos

Após termos identificado os objetos, suas características, seus relacionamentos com outros objetos e seu comportamento, poderemos aplicar uma das várias técnicas de representação de objetos que nos são disponibilizadas. Algumas mais detalhistas, outras mais simplistas, algumas mais formais, outras mais flexíveis.

Sem dúvida, o domínio dessas técnicas ajudará sensivelmente na produção do resultado final. Assim também um artista, se não dominar bem a técnica que utilizará para reproduzir algo que tenha concebido, não obterá como produto algo que realmente demonstre sua intenção. O que é importante, entretanto, é deixar bem claro que:

> *O domínio de técnicas de modelagem é necessário, mas não é suficiente para se produzirem bons modelos.*

Se não privilegiarmos o passo anterior (o entendimento dos conceitos) de nada nos adiantará dominar essa ou aquela técnica! Essa tem sido a grande falha no processo de aprendizado em modelagem de dados. Livros e cursos têm se preocupado excessivamente com as técnicas e explorado de maneira muito superficial o entendimento. Como você já percebeu, já percorremos páginas e páginas deste livro e ainda não foi apresentada

nenhuma das técnicas de modelagem de dados existentes no mercado. Espero que agora você entenda o motivo.

Veremos nos próximos capítulos algumas das técnicas de modelagem de dados existentes, quando você já estiver preparado para elas.

Passo 4: Verificação de fidelidade e coerência

Após terem sido representados novos objetos, através de uma técnica escolhida, devemos verificar se a representação gerada, associada aos objetos já existentes previamente no modelo, continua a formar um conjunto, em sua totalidade, coerente e fiel aos conceitos encontrados.

Isso é necessário pois, muitas vezes, a agregação de novos objetos a um modelo já existente pode vir a mostrar que conceitos anteriores estão distorcidos, ou não são adequados à nova situação. Nesses casos, podemos estar diante de duas situações distintas:

- Os novos objetos agregados ao modelo contêm anomalias e, por isso, destoam do conjunto, devendo então ser remodelados.
- Os novos elementos agregados espelham realmente os objetos observados, o que demonstra a necessidade de tratamento de anomalias nos elementos previamente existentes.

Em ambos os casos, devemos verificar se as falhas, ou anomalias, existem em função de conceitos malformados, pontos de vista equivocados, falha na concepção ou aplicação errada da técnica de representação. Dependendo de onde tenha ocorrido a falha deveremos voltar, mais ou menos, aos passos já executados até o momento para retomar o caminho certo.

Passo 5: Validação

Os procedimentos de validação, que sucedem as verificações feitas no passo anterior, têm papel fundamental no encerramento do processo de modelagem de dados. Será através da validação que obteremos a aprovação formal, ou a indicação de pontos falhos existentes em nossos modelos. Devemos ter em mente, neste instante, duas grandes máximas aplicáveis à modelagem de dados:

> *Não ame seu modelo de dados!*
>
> *Se você acha que seu modelo está bom, é porque talvez ainda não tenha olhado direito!*

Esses dizeres sintetizam tudo aquilo que distingue um bom trabalho de um simples trabalho. Dizem, em outras palavras: altere, mude, melhore, desfaça e, em caso de dúvida, refaça. Somos, apesar de normalmente não admitir, muito paternalistas, ou maternalistas, com nossos modelos. Temos todo um processo de concepção e desenvolvimento e, quando os produzimos, achamos que atingimos a perfeição, nada mais havendo a ser feito, por quem quer que seja, inclusive nós mesmos. Quando alguém começa a questioná-los, a criticá-los e a nos mostrar pontos para ajustes, nos sentimos afrontados.

Você acha isso exagerado?! Procure fazer essa experiência com algum colega seu ou, simplesmente, pare para pensar honestamente em qual foi seu último comportamento quando alguém questionou um de seus últimos modelos.

Por isso acredito que a preparação para atuar em modelagem de dados deve, também, dar muita atenção a esse ponto. Algo que devemos acreditar e ter sempre em mente é que:

> *Se em um processo de validação não conseguirmos descobrir algum ponto falho, alguma anomalia ou algo a ser melhorado, devemos então ficar realmente preocupados pois não fizemos a validação corretamente!*

Certa vez ouvi ou li, em algum lugar, que "em modelagem de dados só podemos ter uma certeza: a de que nosso modelo nunca estará completamente acabado". Acho que esta é, realmente, uma das grandes verdades a serem admitidas quando se fala em validação de modelos. Assim, procure fazer com que o processo de validação de seu modelo seja o mais crítico e rigoroso possível. Coloque seu modelo à prova e veja se ele resiste aos questionamentos. Você só terá a ganhar com isso, conseguindo, o mais rápido possível, antecipar todas as falhas, anomalias ou deficiências que, mais cedo ou mais tarde, acabaria por descobrir, quando então seria tarde, o suficiente, para que os impactos fossem muito maiores.

Outro ponto importante a ser considerado quando se aborda a validação dos modelos de dados é que o grau de conhecimento e domínio das técnicas utilizadas para a representação dos objetos que observamos deve ser homogêneo tanto na equipe de desenvolvimento como no grupo de usuários que participa do processo.

Já presenciei algumas reuniões de validação em que os usuários assistem passivamente à apresentação e às explicações dos modelos de dados que estão sendo validados. Ao final da apresentação, são impelidos a formalmente considerar "validados os modelos", considerando então a fase como encerrada. Mas com que grau de entendimento esses usuários têm, efetivamente, participado do processo? Pouco? Algum? Nenhum?

Caso não haja conhecimento suficiente por parte dos usuários, o processo de validação será meramente formalizador do fechamento de uma etapa. Não terá qualquer finalidade real de apontamento de falhas, omissões, interpretações equivocadas, etc. Logo, a disseminação das técnicas de modelagem de dados deve ser feita também à área usuária e não só ao pessoal envolvido com a área de desenvolvimento de sistemas.

Objetivos do Modelo de Dados

A modelagem de dados tem sido, basicamente, aplicada como meio para obtenção de estruturas de dados que nos levem ao projeto de bancos de dados. Cada vez mais, com a disseminação de sistemas gerenciadores de bancos de dados relacionais (SGBDR), tem sido exigido o formalismo na criação das estruturas de dados a serem implementadas. O efetivo aproveitamento das habilidades e recursos dos SGBDRs tem como ponto fundamental a "obediência" a alguns preceitos relacionais. Melhores ou piores resultados poderão ser obtidos em função de melhores ou piores modelos de dados.

A partir de agora iremos explorar, também, outros pontos que têm sido ignorados quando se trata o assunto modelagem de dados em livros de projeto de bancos de dados. Vamos mostrar, como já havíamos realçado anteriormente, que:

> *Modelagem de dados não é uma ferramenta só para projeto de bancos de dados!*

Sim, mas para que mais poderíamos utilizá-la? As aplicações que iremos apresentar a seguir não são meras especulações ou suposições. São, antes de mais nada, resultado da aplicação das técnicas de modelagem de dados em situações do dia-a-dia. Muitas ocasiões que merecem nosso entendimento, memorização, representação, explicação ou comunicação podem ficar mais fáceis de serem tratadas quando aplicamos alguns conceitos herdados da modelagem de dados.

Dentre essas aplicações aparecem, com maior destaque, não necessariamente em ordem de importância :

- Representar um ambiente observado.
- Servir de instrumento para comunicação.
- Favorecer o processo de verificação e validação.
- Capturar aspectos de relacionamento entre os objetos observados.
- Servir como referencial para a geração de estruturas de dados.
- Estabelecer conceitos únicos a partir de visões diversas.

Representar um Ambiente Observado

Freqüentemente, estamos sendo defrontados com novos ambientes, sejam eles organizacionais, funcionais, pessoais, de hardware, de software, de redes, etc. Esses ambientes possuem características particulares, apresentam aspectos dinâmicos de relacionamento entre seus componentes, possuem regras de funcionamento e operação, entre outras coisas.

Agora pare e lembre! Qual era mesmo a definição de um modelo? "*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 partes.*" Se nos interessa retratar um ambiente (ou sistema, ou conjunto de objetos), ou parte dele, podendo após isso explicar ou transmitir seu comportamento, por que não usaríamos um modelo para isso? E por que não um modelo de dados?

Assim, para retratar com formalismo ou riqueza de detalhes, por exemplo, um ambiente de desenvolvimento de sistemas, com suas equipes disponíveis ou alocadas a projetos e a clientes, com domínios de técnicas de programação e análise diferenciadas e atuando em diferentes ambientes operacionais, poderíamos usar um modelo de dados semântico. Obteríamos resultados tão bons, ou melhores, que utilizando-nos de diagramas, esquemas, tabelas, etc.

O que dizer então dessa técnica aplicada ao mapeamento de regras de funcionamento de uma rede de computadores? E de um ambiente de concessão de créditos bancários? E de um conjunto de lojas de departamento associadas às suas áreas de atuação e fornecedores?

Se você avaliar friamente verificará que, muitas vezes, diagramas, esquemas, descrições e tantos outros meios têm sido utilizados para representar, através de outras técnicas, o mesmo que poderíamos estar representando usando as técnicas de modelagem de dados. Para exemplificar apresentaremos, a seguir, o mesmo ambiente representado por três técnicas diferentes: uma descritiva, uma esquemática e uma através da modelagem de dados.

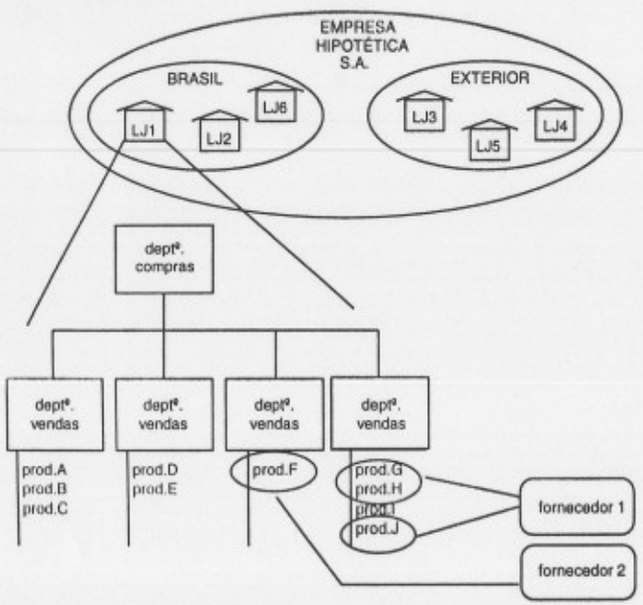
Exemplo:

Utilizaremos para essa demonstração o caso de um hipotético conjunto de lojas de departamento associadas às suas áreas de atuação e fornecedores.
Imagine que você foi chamado a fazer um trabalho de reengenharia organizacional junto a uma cadeia de lojas de departamentos e terá que, inicialmente, fazer o levantamento da situação atual descrevendo-a. Qual poderia ser o resultado desse trabalho? Vejamos:

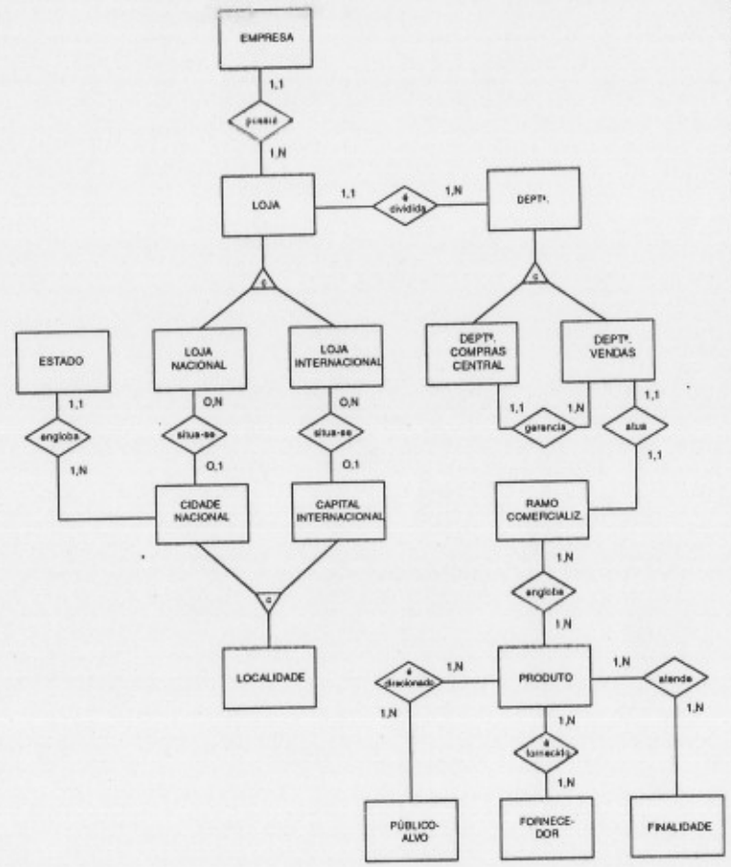
- **Forma 1: Descritiva**

A empresa Hipotética S.A. atua no ramo de lojas de departamentos através de uma cadeia de lojas espalhadas por todo o território nacional e também com lojas nas principais capitais internacionais.
Cada uma das lojas está organizada em uma estrutura hierárquica que é gerenciada por um departamento de compras central, responsável por um conjunto de atividades específicas de negociação e contratação.
Subordinados a esse departamento central estão os demais departamentos de venda que atuam cada um em somente um ramo de comercialização.
Esse ramo de comercialização é definido a partir da agregação de uma série de produtos cuja finalidade ou público-alvo tenham afinidades.
Seus fornecedores são classificados em função dos produtos que representam e dos departamentos de venda afetados por sua linha de produtos.
Muitos dos fornecedores são fornecedores exclusivos de um único departamento de vendas, mas alguns, por sua vez, atendem a diversos departamentos de venda, apresentando produtos específicos para cada um deles.

- **Forma 2: Esquemática**



- **Forma 3: Modelo ados**



Servir de Instrumento para Comunicação

Vários instrumentos têm sido utilizados para favorecer o processo de comunicação, principalmente, no meio empresarial. Hoje, há uma forte preocupação com a preservação do conhecimento adquirido pelos indivíduos durante sua passagem pelas organizações. A rotatividade de pessoal, natural e saudável em muitos casos por oxigenar as organizações, não pode ser motivo de perda dos conceitos trazidos, ou obtidos internamente, pelos profissionais que venham a se afastar da empresa. Assim, a ênfase na documentação e normatização tem sido, cada vez mais, crescente.

Os modelos de dados, então, podem ser aplicados como meio para facilitar o entendimento e a transmissão de conceitos, especificações e regras. A visão que esses modelos são capazes de repassar, para alguém que

também domine a técnica empregada em sua elaboração, é bastante precisa, concisa, objetiva e com pouca, ou nenhuma ambigüidade. Já o mesmo não podemos falar do processo descritivo, normalmente aplicado a esse fim.

Entretanto, muitas pessoas têm se mostrado céticas quanto à possibilidade de aplicação abrangente desse tipo de técnica ao ambiente empresarial argumentando que "esta é uma técnica para a área de informática e não para outras áreas da empresa". Para refutar essa opinião, podemos utilizar o seguinte argumento:

Há alguns anos, no início dos anos 80, quando ainda se utilizava o FORTRAN como linguagem de programação nas universidades, grande parte dos métodos de aprendizado de lógica de construção de programas era feita através de fluxogramas. Esses diagramas pareciam servir a uma finalidade bastante específica: construir programas. Diagramas que ficavam restritos ao mundo dos programadores de computador, como se pensava na época. Mas o que aconteceu com o passar do tempo? Para que finalidade tem se aplicado essa técnica nos dias de hoje? Preste atenção e você verá manuais de operação de fornos de microondas, videocassetes, impressoras, máquinas de lavar pratos e tantos outros explicando o modo de manuseio desses aparelhos através de fluxogramas! E poucas pessoas têm achado isso mais confuso do que páginas e páginas de textos explicativos!

Os mesmos fluxogramas têm sido aplicados, por algumas empresas, como meio de documentação dos procedimentos executados por uma determinada área durante a busca de certificação para ISO-9000. E, nesse caso também, poucos têm achado que esta é uma técnica confusa ou que seja voltada à construção de programas.

Com isso, podemos perceber que a aplicação de técnicas de representação gráfica, que podem parecer voltadas ao ambiente de informática, tal como a modelagem de dados, a finalidades que não sejam propriamente da área de informática, não só é uma possibilidade como também é uma tendência. Assim também outras técnicas de diagramação, como diagramas de fluxo de dados (DFD), diagramas de transição de estado (DTE), entre outros, podem ser aplicadas, de maneira bastante benéfica, a outras finalidades que não a de construção de sistemas de informação.

Favorecer o Processo de Verificação e Validação

Associando as justificativas citadas, veremos que se buscamos um meio de representar (ou modelar) um ambiente, e se podemos disseminar o conhecimento das técnicas aplicadas a esse fim para pessoas envolvidas com o processo mas que não sejam da área de informática, teremos então obtido um meio facilitador para verificação e validação do nosso entendimento sobre determinado assunto.

Teremos a chance de confrontar os conceitos que obtivemos pela análise e representação dos objetos que captamos no ambiente observado com os conceitos estabelecidos por cada um dos componentes de uma

equipe de validação. Já mostramos anteriormente que o processo de validação é necessário e valioso e devemos, então, buscar meios para que ele seja feito do modo mais preciso e fiel possível, sem que isso implique intermináveis sessões de trabalho.

A objetividade, precisão e concisão conseguidas através dos modelos de dados podem, e devem, ser buscadas como facilitadores de todo o processo de verificação e validação.

Capturar Aspectos de Relacionamento entre os Objetos Observados

Como veremos, mais a frente, o processo de modelagem de dados é interessante por não se fixar somente aos objetos existentes em um ambiente observado, mas por também dar grande ênfase ao relacionamento entre esses objetos. O mapeamento do tipo de relacionamento existente entre os objetos é capaz de nos mostrar características tais como: regras de existência desses objetos, regras de evolução, regras de preservação, associações possíveis, associações proibidas, número de participantes nas associações de objetos, grau de hierarquia existente entre os objetos, entre outras.

Tendo-se um meio para mapear os objetos, e podendo utilizar-se desse mesmo meio para mapear os relacionamentos entre esses objetos, pode-se atingir com maior grau de confiabilidade uma representação única que seja consistente na representação de diferentes aspectos de um mesmo ambiente.

Servir como Referencial para a Geração de Estruturas de Dados

Como vimos anteriormente a representação ou documentação dos objetos e relacionamentos existentes em um ambiente observado pode ser uma necessidade associada a qualquer outra finalidade que não a de construção de um sistema de informações.

Imagine que você necessite, em um dado instante, representar a estrutura de atendimento de uma rede de fornecedores de autopeças para as fábricas e lojas de peças de reposição espalhadas nas diversas regiões do país. Imagine ainda que cada região tem escritórios centrais com equipes próprias mas que podem atender também em casos de emergências a clientes de outras regiões. Como você mapearia todos esses objetos e relacionamentos, suas regras, participantes, exceções, etc.? Como já vimos, a modelagem de dados seria uma boa opção para obter e representar todos esses objetos. Se você a utilizasse, obteria um mapeamento bastante apropriado para conter todos esses detalhes.

Mas, e se você decidir, futuramente, implantar sistema informatizado para manter todas estas informações de modo automático? Qual seria seu ponto de partida? Como construiria seu banco de dados? Caso tivesse realizado, em algum instante, um trabalho de modelagem de dados, mesmo

que não visando a qualquer finalidade associada à área de inf[illegible]ática, bastariam apenas algumas complementações e seu banco de dados estaria quase pronto.

Isso demonstra que, além de uma ferramenta apropriada para diversas finalidades fora da informática, também se traduz em investimento qualquer esforço realizado para obtenção de modelos de dados, pois estes, a qualquer instante, podem acelerar sensivelmente o processo de obtenção de estruturas de dados, ou de bancos de dados.

2

Níveis de Modelagem

Definição de Níveis

Durante o ciclo de desenvolvimento de sistemas, convencional ou orientado a objetos, os modelos de dados passarão por níveis distintos, cada qual com suas características e particularidades.

A estratégia de utilização de diferentes níveis de projeto e representação dos modelos de dados tem origem junto ao grupo ANSI-X3-SPARK ainda na década de 1970. Já naquele instante, com a disseminação dos primeiros sistemas gerenciadores de bancos de dados, começavam a surgir dificuldades quanto à implementação de bases de dados e, paralelamente, a busca de padrões que pudessem regrar o ambiente.

O padrão básico de definição e especificação de parâmetros e elementos que compunham um banco de dados, incluindo os aspectos conceituais, lógicos e físicos, estabelecia o uso de *schemas* (esquemas, como alguns tradutores têm preferido). Esses *schemas*, na verdade, eram unidades de especificação que serviam de mapeamento para as estruturas de um banco de dados. Esse mesmo conceito continua sendo aplicado até hoje na implementação dos mais recentes SGBDs.

As dificuldades existentes, antes do estabelecimento da arquitetura de três níveis de *schema*, residiam basicamente em dois pontos:

1. Um mesmo modelo de dados, concebido para uma aplicação, ao necessitar ser implementado em diferentes SGBDs exigia a geração de diferentes schemas com transformações significativas do próprio modelo original. Isso ocorria pois cada schema dependia das características de implementação de cada software. Assim, se a implementação fosse feita em um SGBD hierárquico ou em um SGBD relacional teríamos schemas bastante particulares e reflexos bastante significativos no modelo de dados para possibilitar o estabelecimento das estruturas necessárias entre os dados.

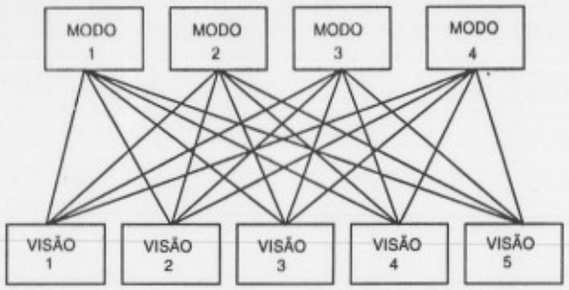
2. A implementação de um mesmo schema em um mesmo SGBD poderia ser percebida, ou disponibilizada ao usuário final, por diferentes visões (*views*). Estas visões seriam dependentes de

restrições de acesso, limitações físicas, características do ambiente de manipulação, etc.

Com essa realidade, um mesmo modelo de dados, gerado para uma aplicação, poderia necessitar de um grande número de diferentes schemas para comportar variações de modos de implementação e de visões externas a serem disponibilizadas aos usuários. Para ser mais exato, esse número poderia ser obtido através da seguinte expressão:

Qtde. de schemas para uma aplicação (conceitual)
=
Qtde. de modos de implementação
x
Qtde. de diferentes visões

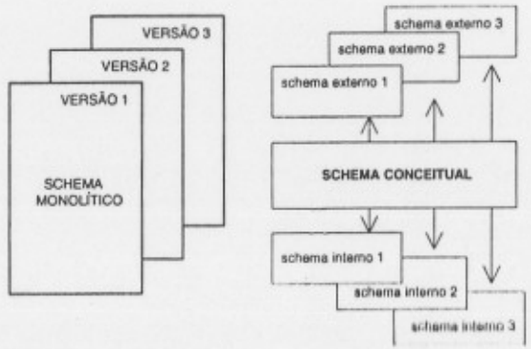
Se imaginarmos, por exemplo, uma situação em que um modelo de dados conceitual original precisasse vir a ter seu schema definido para quatro diferentes modos de implementação convivendo com cinco diferentes visões teremos:



Cada uma das 20 interligações mostradas representa uma possível combinação de características específicas de um modo de implementação e de uma visão a ser disponibilizada ao usuário. Para cada uma delas deveríamos redefinir nosso schema e nosso modelo, onde teríamos, ao final, exatos 20 schemas, e modelos, sendo mantidos para possibilitar a implementação flexível de nosso banco de dados. Você consegue imaginar essa situação em nossos dias se algo não tivesse sido feito até agora para mudar essa situação?

As dificuldades advindas da administração e manutenção de toda essa variedade de modelos fizeram com que o grupo ANSI-X3-SPARK propusesse um padrão onde a idéia básica consistia na definição de níveis para a definição de schemas associados a um modelo de dados. Cada um desses

níveis deveria isolar as características específicas que lhe diziam respeito em um schema próprio. Sintaxes e modos de tratamento seriam definidos para cada um dos níveis e haveria grande isolamento entre eles, fazendo com que mudanças em um nível causassem nenhum ou pouco impacto nos demais. Os schemas de cada um dos níveis seriam então combinados entre si para que pela utilização combinada pudessem vir a formar o conjunto de especificações antes obtidos em um só schema.



antes do padrão ANSI-SPARK — após o padrão ANSI-SPARK

Assim, a quantidade final de schemas, e modelos, necessários a serem mantidos para uma aplicação modelada (schema conceitual) passa a ser não mais calculado pelo produto da quantidade de modos de implementação (schemas internos) *versus* a quantidade de visões a serem disponibilizadas ao usuário final (schemas externos) mas pela sua somatória.

Qtde. de schemas para uma aplicação (conceitual)
=
Qtde. de modos de implementação
+
Qtde. de diferentes visões

No exemplo anteriormente dado, uma aplicação poderia vir a ter quatro modos de implementação e cinco visões. Pelo processo anterior, necessitávamos manter 20 schemas distintos. Agora, baseando-se em uma arquitetura de três níveis de schema teremos somente nove, ou seja, um ganho bastante significativo.

Esse tipo de abordagem passou a ser definido como "arquitetura de três níveis de schema", representando a existência de um nível de schema conceitual, um nível de schema interno e um nível de schema externo. Veremos mais a frente que esse padrão passou a ser bastante compatível

com a proposta dos modelos conceitual, lógico e físico apregoados pela abordagem entidade-relacionamento (E-R) de Peter Chen.

A abordagem E-R defende, basicamente, a elaboração de um modelo que represente os objetos observados, e seus relacionamentos, independentemente de preocupações com implementações lógicas ou físicas. Entende que os aspectos lógicos e físicos devem ser agregados *a posteriori*, pois são alheios à estrutura inerente dos dados observados em um ambiente ou conjunto de objetos. É, portanto, uma técnica que tem total aderência aos padrões definidos na arquitetura de três níveis de schema do grupo ANSI-X3-SPARK e, através de sua aplicação, nos levará, durante o ciclo de desenvolvimento, à convivência com três tipos distintos de modelos de dados: o modelo conceitual de dados (MCD), o modelo lógico de dados (MLD) e o modelo físico de dados (MFD).

Modelo Conceitual de Dados (MCD)

Define-se como modelo conceitual aquele em que os objetos, suas características e relacionamentos têm a representação fiel ao *ambiente observado*, independente de limitações quaisquer impostas por tecnologias, técnicas de implementação ou dispositivos físicos. Nesse modelo, devemos representar os conceitos e características observados em um dado ambiente, voltando-nos simplesmente ao aspecto conceitual.

Esse modelo deve ser o modelo a ser utilizado para o nível de conversação, entendimento, transmissão, validação de conceitos, mapeamento do ambiente, etc. Nesse nível devem ser ignoradas quaisquer particularidades de implementação, bem como desconsiderada qualquer preocupação com qual será o modo de implementação futura. Assim, nosso modelo conceitual de dados (MCD) permanecerá imutável tanto se vier a ser implementado em um SGBD relacional, como em um SGBD hierárquico.

Na verdade, essa estabilidade é o grande diferencial na utilização de um modelo conceitual de dados. Devemos *derivar* diferentes estruturas de implementação a partir de um mesmo modelo conceitual, e não criá-las e mantê-las separadamente. Assim, a partir de um MCD poderemos, a qualquer instante, derivar um modelo para implementação em SGBD relacional. Se, futuramente, nossa necessidade não for mais de um modelo relacional, descartaremos o modelo derivado, voltaremos ao modelo conceitual original, aplicaremos outras regras de derivação e poderemos ter um outro tipo de modelo novamente derivado.

Junto às metodologias de desenvolvimento de sistemas e implementado por ferramentas CASE, esse modelo surgirá sempre associado às fases de análise e nunca às fases de projeto. Isso porque nesse nível de modelagem de dados ainda não temos interesse em modos de implementação ou outras características de construção. Mais especificamente no caso das ferramentas CASE, modelos dessa natureza se destacarão no segmento de UPPER-

CASE, ou seja, naq[illegible]s que atuam em etapas que antecedem o próprio ciclo de desenvolvimento de sistemas. Assim, ferramentas CASE que atendam às etapas de planejamento e análise estarão implementando esse tipo de modelo durante seu trabalho de diagramação.

No caso de projeto de bancos de dados, deveremos estar atentos a não construirmos nossos modelos já como modelos lógicos de dados. Apesar de isso ser possível, e até muitas vezes induzido por algumas ferramentas CASE, devemos, mesmo que num primeiro instante não nos pareça necessário, construir os modelos conceituais de dados que antecedem a fase de projeto dos BDs. Caso passemos direto à construção dos modelos lógicos de dados estaremos nos limitando a utilizar no processo de modelagem somente os recursos disponibilizados por uma ou outra tecnologia de implementação.

Outro ponto importante, que justifica o investimento na criação do modelo conceitual de dados, é que o esforço feito para sua geração é igual ou, muitas vezes, menor que o realizado para gerar o modelo lógico de dados. Não precisaremos, durante sua elaboração, estar preocupados com, além de conhecer, entender e representar os objetos observados, respeitar as regras e limitações dessa ou daquela tecnologia. Poderemos nos concentrar puramente no aspecto conceitual obtendo mais e melhores detalhes sobre os objetos que nos interessam naquele instante.

Se desejarmos, por outro lado, aplicar a modelagem de dados a outras finalidades que não a de projetar bancos de dados, obrigatoriamente deveremos nos utilizar de modelos conceituais de dados. Regras e limitações de tecnologia não nos interessam nesse caso e, por isso, deverão ser descartadas.

Modelo Lógico de Dados (MLD)

Define-se como modelo lógico de dados (MLD) aquele em que os objetos, suas características e relacionamentos têm a representação de acordo com as regras de implementação e limitantes impostos por algum tipo de tecnologia. Essa representação, por sua vez, é independente dos dispositivos ou meios de armazenamento físico das estruturas de dados por ela definidas.

Esse modelo deve ser o modelo elaborado respeitando-se e implementando-se conceitos tais como chaves de acesso, controles de chaves duplicadas, itens de repetição (arrays), normalização, ponteiros, headers, integridade referencial, entre outros. Estas são preocupações e necessidades somente relevantes ao MLD, jamais ao MCD; portanto, devem ser abordadas somente neste instante.

Durante o processo de desenvolvimento de sistemas e implementado por ferramentas CASE, esse modelo estará sempre associado à fase de projeto. Muitas ferramentas CASE que atuam somente no nível LOWER-CASE implementam somente esse tipo de modelo, não favorecendo o

desenvolvimento de um modelo conceitual ou tentando fazê-lo através do próprio modelo lógico. O modelo lógico é bastante voltado à implementação sob o ponto de vista de sistemas de informação, não sendo, portanto, indicado para aplicações que extrapolem a área de informática, como já vimos.

A obtenção do MLD se dará pela aplicação de regras de derivação sobre um MCD já construído. Essas regras serão devidamente tratadas nos próximos capítulos. Este será o tipo de abordagem de obtenção do modelo lógico que defenderemos, baseados no fato de que, como pré-requisito, já estabelecemos a necessidade de elaboração de um modelo conceitual.

Muitos autores e metodologias que não prestigiam o papel do modelo conceitual de dados têm definido que o método para a obtenção do modelo lógico é o próprio processo de modelagem criativa. Através desse outro enfoque, partimos para a observação dos objetos e já no primeiro instante quando uma representação seja possível utilizamos os conceitos e restrições do modelo lógico escolhido para construir, então, o suposto modelo de entidade-relacionamento. Devemos lembrar que esse modelo não estará somente preocupado com a representação dos objetos observados, mas também com a representação de conceitos necessários à implementação tais como chaves, métodos de acesso, formatos de campos, etc. Isso significa, em última análise, que do ponto de vista formal de definição da abordagem entidade-relacionamento, proposta por Chen, esse modelo não se enquadra fielmente como um modelo E-R.

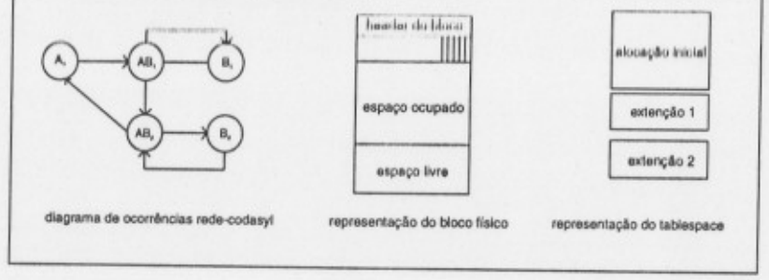
Modelo Físico de Dados (MFD)

Define-se como modelo físico de dados aquele em que a representação dos objetos é feita sob o foco do nível físico de implementação das ocorrências, ou instâncias das entidades e seus relacionamentos. O conhecimento do modo físico de implementação das estruturas de dados é ponto básico para o domínio desse tipo de modelo.

Cada diferente SGBD poderá definir um diferente modo de implementação física das características e recursos necessários para o armazenamento e manipulação das estruturas de dados. Em alguns casos, um mesmo SGBD em diferentes ambientes de sistema operacional poderá ter diferentes métodos de armazenamento e manuseio de suas estruturas de dados. Isso é fácil de ser constatado quando analisamos um SGBD que tenha sua implementação, por exemplo, em ambiente DOS, UNIX e VMS. Em cada um desses ambientes, existirão diferentes estruturas de armazenamento, endereçamento, acesso e alocação física. Isso significa que, em alguns casos, um mesmo modelo lógico poderá estar mapeado de diferentes modos em cada um dos sistemas operacionais.

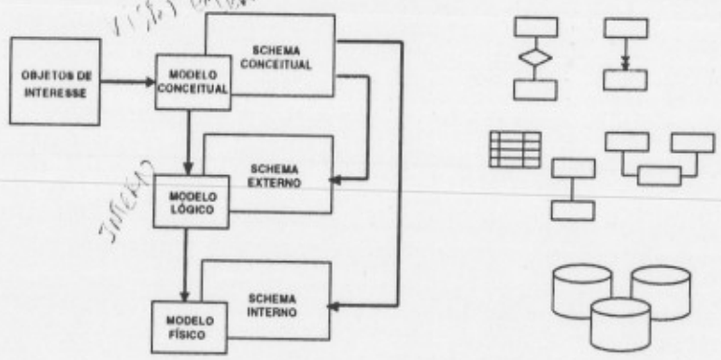
No modelo físico podem ser incluídos dois níveis de representação. O primeiro deles diz respeito às ocorrências ou instâncias, seus relacionamentos e disposição física de elementos. O outro diz respeito à alocação de espaços físicos nos diversos níveis de agrupamento possíveis: tabelas (arquivos), blocos, linhas (registros) e colunas (campos).

Exemplo:



Integração da Arquitetura de Três Níveis com a Abordagem E-R

Pelo que já pudemos definir até o momento, pode ser verificada que uma integração entre a abordagem de modelos conceitual, lógico e físico e a proposta do grupo ANSI-X3-SPARK é possível se forem estabelecidos os níveis de projeto conforme o diagrama a seguir:



Assim poderá ser constatado que, partindo-se de um mundo observado, composto por seus objetos e relacionamentos, poderemos definir um modelo independente de tecnologia (relacional, rede, hierárquica). Este modelo poderá ser derivado para um modelo lógico que por sua vez será dependente de modelos físicos de implementação.

A cada um desses níveis de modelagem serão associadas técnicas de representação gráfica e métodos e de especificação de schemas. Este tipo

de especialização para cada um dos níveis fará com que sejam necessários conhecimentos específicos para o entendimento e manipulação dos elementos envolvidos em seus escopos.

Nos próximos capítulos estaremos desenvolvendo essas habilidades através do tratamento de cada um dos níveis em profundidade.

3

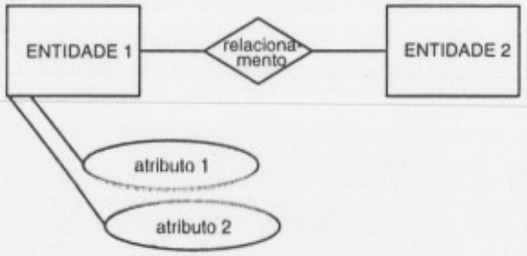
Modelo E-R

A Abordagem Entidade-Relacionamento

Em março de 1976, Peter P. Chen publicou um trabalho intitulado "The Entity-Relationship Model: Toward the unified view of data",[1] no qual definia uma possível abordagem para o processo de modelagem dos dados. Esse trabalho, após sua divulgação e ampla aceitação, passou a ser considerado como um referencial definitivo para o processo de modelagem de dados. A abordagem entidade-relacionamento é composta de uma técnica de diagramação e de um conjunto de conceitos que devem ser entendidos e respeitados.

A técnica de diagramação é bastante simples e serve como meio de representação dos próprios conceitos por ela manipulados. Utiliza, basicamente, um retângulo para representar as entidades, um losango para representar os relacionamentos e balões para indicar e alocar os atributos.

Abordagem Entidade-Relacionamento



O Embasamento para o Modelo E-R

A proposta original de Chen, que se estabeleceu e mantém-se extremamente atualizada até os dias de hoje, baseia-se em um princípio que a torna ao mesmo tempo completa e inquestionável: a formalização do óbvio. Aparentemente, esse pode parecer um meio um tanto simplista de se definir todo o embasamento conceitual no qual se baseia o modelo de entidade-relacionamento. Entretanto, é justamente por apresentar essa simplicidade e

objetividade que a abordagem entidade-relacionamento possui ita flexibilidade e adaptabilidade. Ao contrário do que possa parecer, essa é sua maior qualidade.

Ao observarmos o modelo proposto por Chen, podemos verificar que basicamente o que ele define, em outras palavras, é o que chamaremos de *A Lei do Mundo*:

> *A Lei do Mundo*
> *O mundo está cheio de coisas que possuem características próprias e que se relacionam entre si.*

Extremamente simples, mas, ao mesmo tempo, extremamente correta. Ninguém, em tempo algum, será capaz de demonstrar algo contrário a isso, seja no mundo microscópico ou macroscópico. Chen, é bem verdade, procurou não usar o termo "coisas", o que tornaria sua proposta muito informal, preferindo substituí-lo por "entidades", que denota de modo igual os objetos ou elementos observados.

Chen teve a chance de observar e enunciar aquilo que todos já sabiam mas estavam incapacitados de aplicar de modo concreto ao processo de modelagem: "Observamos, manipulamos, reproduzimos ou simulamos coisas, seus relacionamentos e características e, portanto, assim devemos representá-las!"

Reconhecimento dos Elementos no Ambiente Observado

Utilizando se da "Lei do Mundo", que representa o embasamento teórico para o modelo de entidade-relacionamento, iremos procurar identificar e conceituar alguns elementos de nosso interesse: as entidades, os relacionamentos e os atributos. Futuramente, veremos que outros elementos poderão vir a ser identificados e aplicados ao modelo E-R. Esses novos elementos farão parte de extensões do modelo proposto por Chen e estarão sendo utilizados com a finalidade de agregar mais semântica ao modelo a ser construído. Esses novos elementos servirão, também, de meio para uma representação voltada ao ambiente orientado a objetos, dentro do enfoque de modelagem de informação.

Como já definimos, o processo de modelagem pressupõe a existência de um objeto, ou ambiente, a ser observado. A fase de especificação de requisitos e o próprio processo de observação acabarão por definir que nível de abrangência estará sendo utilizado. Entretanto, independentemente de estarmos focados na nossa visão sobre um objeto ou sobre todo um ambiente estamos sendo confrontados com os mesmos elementos de interesse no processo de modelagem.

Dentro do enunciado que definimos para a "Lei do Mundo", passamos então a identificar os seguintes pontos de interesse:

1. "O Mundo Está Cheio de Coisas"

Define que dentro de um universo observado (O Mundo) estaremos reconhecendo objetos (Coisas). Esses objetos estarão sendo percebidos como elementos individualizados mas, ao mesmo tempo, poderão ser enquadrados em um conjunto ou categoria em função de suas semelhanças. Esse conjunto, futuramente, dependendo da abordagem dada ao processo de modelagem, poderá vir a ser definido como *entidade* ou *classe*.

Exemplo:

Ao observarmos um ambiente de produção de uma fábrica, nos defrontaremos com:

- Máquinas de produção de peças.
- Funcionários operadores dessas máquinas.
- Conjuntos de ferramentas utilizadas para operar as máquinas.
- Procedimentos de operação a serem executados.
- Procedimentos de verificação de qualidade das peças produzidas.
- Máquinas recuperadoras de peças.
- Funcionários responsáveis pela verificação da qualidade das peças.
- Peças produzidas nos mais diversos formatos.

Dentre esses elementos observados, poderemos perceber a existência de vários conjuntos distintos de elementos, entre eles:

- **MÁQUINA.** Que representará todas as máquinas observadas, independentemente de suas diferentes aplicações, ou de quem as opere.
- **FUNCIONÁRIO.** Que representará todos os funcionários, independentemente de suas funções ou das máquinas com as quais esteja envolvido.
- **FERRAMENTA.** Que representará todas as ferramentas existentes, independentemente de suas finalidades ou uso.
- **PEÇA.** Que representará todos os tipos de peças produzidos em qualquer tipo de máquina, por qualquer funcionário e por qualquer processo.

2. "Que Possuem Características Próprias"

Define que para os objetos identificados em nosso universo, em nível individual, estaremos percebendo características comuns que os levarão a ser enquadrados em conjuntos particulares. Essas características são inerentes a todos os objetos de um mesmo conjunto e assim deverão ser representadas. Todos os objetos (elementos individualizados) que possuam

uma mesma característica serão denominados, simplesmente, pelo nome do conjunto ao qual pertençam.

Exemplo:

"Conselho Regional de Medicina", "Conselho Regional de Odontologia" e "Ordem dos Advogados do Brasil" serão objetos, ou elementos individualizados, pertencentes ao conjunto denominado *ÓRGÃO NORMALIZADOR*.

"João", "Pedro", "José", "Maria" e "Ana" serão objetos, ou elementos individualizados, pertencentes ao conjunto denominado *FUNCIONÁRIO*.

"General Motors do Brasil" não se enquadrará nem como ÓRGÃO NORMALIZADOR nem como FUNCIONÁRIO, pois possui características diferenciadas que a fazem pertencer ao conjunto que engloba os objetos do tipo *FABRICANTE*.

3. "E Que Se Relacionam entre Si"

Define que entre os objetos, ou entre os conjuntos, existirão inter-relacionamentos (Se Relacionam) particulares estabelecidos em função de suas próprias características. Esses relacionamentos são inerentes aos objetos de um dado conjunto e assim deverão ser representados. Os relacionamentos poderão estar sendo observados entre elementos individualizados de diferentes conjuntos ou entre elementos individualizados de um mesmo conjunto.

Exemplo:

"João é credenciado pelo Conselho Regional de Medicina" é um relacionamento entre elementos individualizados (João e Conselho Regional de Medicina) de diferentes conjuntos (FUNCIONÁRIO, ÓRGÃO NORMALIZADOR).

"João é substituto de Pedro" é um relacionamento estabelecido entre elementos individualizados (João e Pedro) de um mesmo conjunto (FUNCIONÁRIO).

"Conselho Regional de Odontologia reporta-se ao Conselho Regional de Medicina" é um relacionamento entre elementos individualizados (Conselho Regional de Odontologia e Conselho Regional de Medicina) de um mesmo conjunto (ÓRGÃO NORMALIZADOR).

Partindo-se desse ponto e aplicando-se o princípio da abstração, poderemos chegar à elaboração dos conceitos necessários dentro da abordagem entidade-relacionamento:

Conceito Básico na "L... do Mundo"	Abordagem do Modelo E-R (Modelagem de Dados)	Abordagem Orientada a Objetos (Modelagem da Informação)
Conjunto de objetos, ou elementos individualizados, semelhantes	Entidades	Classes
Individualização de objetos ou elementos	Instâncias ou Ocorrências	Objetos
Características próprias	Atributos	Atributos
Inter-relacionamento de objetos	Relacionamentos	Associações

Entidades

Identificação de Objetos (Entidades ou Classes)

Dentre as diversas estratégias para reconhecimento e identificação de objetos para sua posterior alocação em conjuntos (entidades ou classes), vamos apresentar a proposta por Shlaer & Mellor no livro *Object-Oriented Systems Analysis, Modelling the World in Data*.

Essa estratégia define que deveremos procurar reconhecer objetos, ou elementos individualizados, através da observação de cinco grandes grupos de elementos:

- As coisas tangíveis.
- As funções exercidas por elementos.
- Eventos ou ocorrências.
- Interações.
- Especificações.

Shlaer & Mellor procuram deixar claro que essa abordagem não necessariamente implica a definição formal de um processo de classificação de objetos. Para fins didáticos e para nivelamento com estratégias de outros autores, chamaremos essa separação em grupos de *classificação*.

Outros autores propõem outras classificações possíveis e aplicáveis de maneira similar. Basicamente, o resultado final da aplicação de uma ou outra estratégia acabará por nos levar ao reconhecimento dos mesmos elementos. Na verdade, após alguma experiência com o processo de modelagem, esse reconhecimento passa a ser intuitivo, não necessitando mais de uma busca de enquadramento em uma classificação formal qualquer.

As classificações propostas têm como objetivo servir como um meio de discernimento para casos não facilmente enquadráveis ou mesmo para verificação e validação do processo de modelagem.

Abordaremos cada um dos cinco grandes grupos, procurando caracterizá-los e exemplificá-los. Ao final dessa exposição, você terá toda a informação necessária para identificar e agrupar os objetos em seus conjuntos.

Coisas Tangíveis

Como o próprio termo define, tangível é tudo aquilo que pode ser tocado. Assim, o grupo de "coisas tangíveis" engloba todos os elementos que tenham existência concreta, que sejam manipuláveis, que possam ser tocados, que ocupem um lugar no espaço, ou qualquer outro tipo de visão que nos leve a considerá-lo como fisicamente existente.

Exemplo:

Um avião a jato, um cavalo puro-sangue, um livro de ficção, uma garrafa térmica, uma chave da porta de entrada, o lápis preto, sua borracha usada, os computadores da rede, o disquete de demonstração, um quadro-negro, sua máquina de escrever, o cachorro *dobermann* do vizinho, a mesa de trabalho, o telefone sobre a mesa, a mala que transporta seus pertences, as camisas mandadas à tinturaria, os dois vidros de compota que quebraram, uma carteira de dinheiro perdida, sua agenda de compromissos, os elefantes do circo, seu automóvel novo, etc.

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 à formação de diferentes conjuntos.

Com um nível de abstração maior, ou seja, desconsiderando maiores detalhes, poderíamos dentre os objetos apresentados definir os seguintes conjuntos:

Conjunto (entidade/classe)	Objetos Que Pertencem ao Conjunto
MEIO DE TRANSPORTE	o avião, o automóvel
ANIMAL	o cavalo, os elefantes, o cachorro
UTENSÍLIO DOMÉSTICO	a garrafa, a mesa, o telefone, os vidros
UTENSÍLIO ESCOLAR	o livro, o lápis, a borracha, o quadro, o disquete
EQUIPAMENTO	os computadores, a máquina de escrever
PERTENCE PESSOAL	a chave, a mala, as camisas, a carteira de dinheiro, a agenda

Outro nível de abstração, mais generalista ainda, poderia simplesmente ter definido os seguintes conjuntos:

Conjunto (entidade/classe)	Objetos Que Pertencem ao Conjunto
UTENSÍLIOS	a garrafa, a mesa, o telefone, os vidros, o livro, o lápis, a borracha, o quadro, o disquete, a chave, a mala, as camisas, a carteira de dinheiro, a agenda
ANIMAIS	o cavalo, os elefantes, o cachorro
PRODUTOS	o avião, o automóvel, os computadores, a máquina de escrever

Por outro lado, um nível de abstração extremamente especializado poderia ter observado vários aviões, vários automóveis, vários cavalos e assim por diante e ter criado os seguintes conjuntos:

Conjunto (entidade/classe)	Objetos Que Pertencem ao Conjunto
AVIÃO	todos os exemplares de aviões observados individualmente
AUTOMÓVEL	todos os exemplares de automóveis observados independentemente de sua marca, modelo ou tipo
CAVALO	todos os exemplares de cavalos observados, independentemente de sua raça

Com esses exemplos, pode-se verificar que mesmo dentro do grupo de COISAS TANGÍVEIS, que será aquele que teremos maior facilidade em identificar, existem possíveis níveis de abstração a serem aplicados na definição dos conjuntos de nosso interesse (entidades ou classes).

Esse nível de abstração, que nos levará a diferentes níveis de agregação dos objetos, deverá ser definido em função da abrangência definida no processo de especificação dos requisitos.

Veremos, mais a frente, que conjuntos que demonstrem generalizações de objetos, tal como UTENSÍLIOS que define no último exemplo qualquer tipo de utensílio seja ele doméstico, escolar ou pessoal, podem vir a ser melhor explorados com a utilização de recursos adicionais junto aos elementos básicos da abordagem entidade-relacionamento.

Funções

Muitas vezes percebemos os objetos, ou elementos individualizados, através da função por eles exercida. Nesses casos, um objeto que, muito provavelmente, estaria sendo caracterizado como uma COISA TANGÍVEL passará a ser enquadrado na classificação de FUNÇÕES.

Por FUNÇÕES, definiremos todo o tipo de papel, atribuição, classificação, capacitação, ou outra característica qualquer que, para um dado ele-

mento, especifique não sua existência mas sua atuação no ambi[illegible]e em que está inserido.

Exemplo:

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 do hotel, um médico pediatra, a seção de despachos de material, o gerente do hotel, um paciente que é atendido, os alunos de uma escola, etc.

Pode-se perceber que, nesse caso, existem FUNÇÕES sendo desempenhadas por pessoas (o médico, o autor, etc.) e por órgãos funcionais (o departamento de compras, a gerência de suporte técnico, etc.). Esse tipo de visão deve ser mantida em mente, pois será bastante freqüente a aparição desses tipos de objetos.

Também com relação à formação de conjuntos, nesse caso, deveremos levar em consideração o grau de abstração definido na especificação dos requisitos da modelagem. Se essa abstração for, por exemplo, de alto nível, poderemos definir os seguintes conjuntos:

Conjunto (entidade/classe)	Objetos Que Pertencem ao Conjunto	COISAS TANGÍVEIS Mapeadas como FUNÇÃO
ESPECIALISTA	o médico cirurgião, o médico pediatra, o engenheiro naval, o autor de um livro	PESSOA
ORGANIZAÇÃO	um departamento de compras, a gerência de suporte técnico, a seção de despachos de mercadorias	ÓRGÃO FUNCIONAL
ATENDENTE	o professor de inglês, o gerente do hotel, a recepcionista do hotel	PESSOA
CLIENTE	os alunos, o paciente	PESSOA

Nesse caso, vimos que grande parte das funções detectadas foi relativa a papéis exercidos por PESSOAS. Essas pessoas, caso tivessem sido mapeadas como COISAS TANGÍVEIS precisariam, eventualmente, em algum instante, ser qualificadas como médicos ou pacientes, professores ou alunos. Desse modo, a explicitação de suas funções já num primeiro instante da modelagem pode ser vista, conceitualmente, como um bom recurso para a obtenção de maior semântica aos modelos.

Caso no exemplo anterior procurássemos utilizar uma abstração de nível mais baixo, poderíamos ter definido muito mais funções. Nesse caso,

teríamos muito mais [illegible]nentos semânticos em nosso modelo conceitual. Devemos, entretanto, estar atentos ao nível de abstração desejado para não poluir nosso modelo com elementos desnecessários. Um exemplo do que poderíamos ter obtido se nosso nível de abstração fosse bastante baixo seria:

Conjunto (entidade/classe)	Objetos Que Pertencem ao Conjunto	COISAS TANGÍVEIS Mapeadas como FUNÇÃO
MÉDICO	o médico cirurgião, o médico pediatra	PESSOA
ENGENHEIRO	o engenheiro naval	PESSOA
ÁREA DE COMPRAS	o departamento de compras	ÓRGÃO FUNCIONAL
PROFESSOR	o professor de inglês	PESSOA
AUTOR	o autor do livro	PESSOA
ÁREA DE SUPORTE	a gerência de suporte	ÓRGÃO FUNCIONAL

Eventos ou Ocorrências

Alguns objetos, ou elementos individualizados, só conseguem ser percebidos, ou caracterizados, enquanto uma certa ação se desenrola. Em outras palavras, quando alguma ação, ou fato, acontece conseguimos definir características que os tornam materializáveis. Enquanto programado, durante sua execução ou após encerrado, esse elemento caracteriza-se como um evento ou ocorrência ao qual podemos fazer alguma referência.

Exemplo:

Um vôo comercial, um acidente de trânsito, uma apresentação técnica de um fornecedor, uma festa beneficente, uma gincana esportiva, um jogo de futebol, etc.

Em cada um desses casos podemos perceber que o evento, na situação de programado, sendo executado ou realizado possui por si só poucos dados próprios. A maioria dos dados a ele pertencentes são na verdade dados de outros objetos participantes do próprio evento.

Assim, no caso de uma apresentação técnica de um fornecedor, teríamos como dados da própria apresentação:

- Data de realização.
- Horário de início programado.
- Horário de início efetivo.
- Duração prevista.

- Duração efetiva.
- Tema.

Entretanto, outros dados relevantes à apresentação existiriam mas seriam mais fielmente enquadrados se colocados junto aos próprios conjuntos aos quais pertencem e então relacionados com o próprio evento. Esse tipo de abordagem (de relacionamento entre elementos) será mais detalhadamente explorado nos próximos tópicos. O importante, por hora, é identificarmos que outros dados poderão ser definidos como relevantes e agregados ao modelo

Dados Relevantes	Conjunto ao Qual Pertence
nome do apresentador	PESSOA
especialidade do apresentador	PESSOA
telefone para confirmação da presença	FORNECEDOR
endereço da apresentação	HOTEL
nome do produto apresentado	PRODUTO
finalidade do produto apresentado	PRODUTO

Mesmo que a percepção dos conjuntos ideais nos quais os dados relevantes devam ser agregados não seja obtida, teremos através da técnica de normalização meios para realocar os itens de dado que consideramos relevantes em seus devidos lugares. A técnica de normalização será um tópico a ser abordado no projeto lógico. Agora, neste instante, nosso modelo pretende ser um modelo conceitual e, por isso, não requer, ainda, formalismos como a normalização.

Uma consideração importante a ser feita, entretanto, é de que mesmo no nível conceitual devemos buscar, sempre, a definição de conjuntos (entidades/classes) adequados. Ainda muito mais significante isso poderá se tornar se estivermos inseridos em um ambiente de orientação a objetos.

Interações

Os chamados objetos-interação são resultantes da associação de objetos em função de um processo executado. Os objetos-interação passam a existir de maneira individualizada, além dos objetos que participam dessa interação. Pode-se dizer que existe a *materialização* de uma interação entre objetos onde cada um dos objetos constituintes da interação preserva suas características, não sendo impactado por essa materialização.

Esses objetos, em muitos casos, podem ser confundidos, e até mesmo substituídos, por associações (relacionamentos) entre objetos, por coisas tangíveis ou por eventos. Do ponto de vista conceitual, essa substituição mantém todas as características e recursos necessários à futura derivação

das estruturas lógicas e físicas que se façam necessárias. Assim, se você achar mais conveniente não expressar objetos-interação e sim substituí-los por relacionamentos, ou associações, pode fazê-lo sem receios de estar comprometendo a consistência e veracidade de seu modelo.

Exemplo:

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

Perceba que em todos os exemplos apresentados temos interações entre objetos sendo materializadas como novos objetos, mas que também podemos substituí-las por outras representações:

Objeto-Interação	Objetos Que Participam da Interação	Substituição Possível
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"

Especificações

O último grupo de objetos a ser analisado é aquele formado pelos chamados objetos-especificação. Nesse conjunto, aparecem os elementos que definem características de outros objetos. Eles, isoladamente, não representam objetos mas sim especificações que quando aplicadas ou seguidas darão origem a objetos.

Esse grupo, conceitualmente falando, não tem muita significância se analisado isoladamente. Se observarmos do ponto de vista conceitual, os atributos que aparecem como determinantes de um objeto-especificação seriam perfeitamente alocáveis nos próprios objetos que procuram denotar. Desse modo, pode ser considerada válida a supressão desse tipo de objeto no modelo conceitual. Durante a fase de normalização (processo que visa

à obtenção do modelo lógico) acabaremos naturalmente voltando a obtê-los como estruturas de dados independentes.

Os objetos-especificação aparecem normalmente associados ao processo de modelagem com ênfase para o modelo lógico. Como esse tipo de modelo pressupõe um processo de normalização, prévio ou posterior, as pessoas, com o passar do tempo, são induzidas a já conceber seus modelos normalizados. Isso é o que chamamos de *normalizar durante a modelagem*. Entretanto, algo que devemos ter bastante claro quando falamos de modelos conceituais é que o modelo conceitual não precisa, necessariamente, estar normalizado! A normalização é uma exigência somente do processo de modelagem lógica!

Com base nesses pontos, pode-se definir as seguintes opções para o tratamento dos objetos-especificação:

1. Os objetos-especificação podem ser omitidos dos modelos conceituais de dados, desde que seus atributos sejam alocados aos objetos aos quais se referenciam. Durante o processo de modelagem lógica voltaremos a obtê-los e tratá-los como estruturas independentes. Essa abordagem é bastante indicada para o processo de modelagem de dados.
2. Os objetos-especificação podem ser mantidos nos modelos mas acabarão por denotar uma realidade muito próxima do modelo lógico, o que em alguns casos poderá ser prejudicial ao entendimento. No caso da construção de modelos de informação (dentro da abordagem orientada a objetos), essa opção poderá prevalecer, pois, nesse tipo de enfoque, a visão de modelos puramente conceituais não tem sido muito aplicada, apesar de válida. Os modelos construídos dentro do processo de modelagem de informação passam a ter uma tendência a se aproximarem bastante dos modelos lógicos, o que é bastante compatível com essa opção.

Exemplo:

Suponha que você esteja modelando uma fábrica de refrigeradores. Provavelmente entre os objetos lá encontrados teremos:

- MÁQUINAS
- OPERÁRIOS
- FERRAMENTAS
- ACESSÓRIOS
- REFRIGERADORES
- CAMINHÕES TRANSPORTADORES

Imagine que dentre os atributos identificáveis para um refrigerador estejam:

- Sua cor.
- Sua capacidade de armazenagem no compartimento principal.
- A voltagem em que opera.
- O modelo.
- O tempo gasto em sua produção.
- Sua altura.
- Sua largura.
- Sua profundidade.
- A data de sua produção.
- O número de série.
- A data de expiração de sua garantia.

Caso desejássemos, seguindo a opção 1, esses atributos poderiam todos permanecer como atributos de um objeto pertencente ao grupo COISAS TANGÍVEIS, chamado REFRIGERADOR. Assim teríamos:

REFRIGERADOR
cor
capacidade principal
voltagem
modelo
tempo gasto na produção
altura
largura
profundidade
data da produção
número de série
data de expiração da garantia

Seguindo a opção 2 e criando-se um novo objeto pertencente ao grupo dos OBJETOS-ESPECIFICAÇÃO, poderíamos definir a existência dos seguintes objetos e seus atributos:

- Objeto do tipo COISA TANGÍVEL

REFRIGERADOR
cor
modelo
tempo gasto na produção
data de produção
número de série
data de expiração da garantia

- Objeto do tipo ESPECIFICAÇÃO

MODELO DE REFRIGERADOR
capacidade principal
voltagem
modelo
altura
largura
profundidade

Pode se perceber que, em nível de informação, ou de semântica, as duas opções nos levam ao mesmo entendimento sobre o objeto REFRIGERADOR. No primeiro caso, temos todos os atributos alocados ao próprio objeto. No segundo caso, temos os atributos personalizáveis (cor, data de fabricação, etc.) a cada instância do objeto REFRIGERADOR alocados no próprio objeto. Através do atributo *modelo*, podemos obter as demais informações, comuns a todas as instâncias de REFRIGERADOR que se enquadrem em um mesmo modelo, no conjunto de especificações de MODELO DE REFRIGERADOR.

As duas soluções são conceitualmente corretas. A segunda é muito mais orientada ao modelo lógico e poderá parecer melhor principalmente para aqueles que têm pouca experiência no uso de modelos conceituais.

A Dicionarização dos Objetos Modelados

Até o momento, discutimos algumas estratégias e abordagens possíveis para a identificação e caracterização dos objetos que fomos capazes de observar em um universo de investigação. Esses objetos passaram por um processo de abstração de suas características e deram origem a um elemento único (a entidade ou classe) capaz de o representar. Cada um dos

elementos identificados e representados deverá, em um segundo instante, ser definido claramente para que, associando-se seu nome, sua representação e sua definição, sejamos capazes de ter o completo entendimento do conceito que estes procuram transmitir.

Caso tenhamos deficiências em qualquer um desses aspectos (na identificação, na representação ou na definição), poderemos estar inserindo em nosso modelo uma anomalia chamada *ambigüidade*. Por ambigüidade entenda-se: falta de clareza, falta de precisão, incerteza, dúvida, dualidade de interpretação, formação de conceitos equivocados, etc.

Sempre que falo em ambigüidade relembro um fato verídico no qual participei como criador de uma situação bastante ambígua. Foi uma cena que, no instante em que se desenrolava, não parecia conter qualquer ambigüidade (como sempre acontece!), mas que depois, analisada friamente, criou um quadro bastante engraçado. Perceba como foi:

> Há algum tempo, quando estava de plantão em um final de semana, meu bip foi acionado enquanto eu fazia compras em um *shopping center*. Como era um fim de semana em que fazia muito frio, eu estava usando um agasalho de ginástica e uma jaqueta toda branca. A cena que se desenrola, na seqüência, é a seguinte:
>
> 1. O bip toca enquanto estou dentro da loja sendo atendido.
>
> 2. Peço à vendedora para utilizar o telefone por alguns minutos.
>
> 3. Ela me leva até um telefone, junto ao balcão, e aguarda enquanto faço a ligação.
>
> 4. Ligo para o número que me foi passado pelo bip e pergunto:
> — *É da sala da operação?* (referindo-me à sala dos computadores...)
>
> 5. A moça olha para mim um tanto curiosa.
>
> 6. Alguém responde lá do outro lado:
> — *Sim, é da operação. O programa X que estava sendo executado não chegou ao fim de modo normal!*
>
> 7. Continuo com as perguntas:
> — *O que aconteceu? Abortou ou foi morto?*
>
> 8. A moça me olha, espantada com a pergunta!
>
> 9. Alguém responde lá do outro lado:
> — *Abortou. O que devemos fazer? Reprocessar?*

10. Respondo sem grande preocupação:
— *Se abortou não tem problema! Pode deixar assim mesmo. Era só um teste. Podemos fazer outro amanhã. Tchau.*

11. Desligo o telefone e volto às compras com naturalidade.

12. A vendedora fica meio sem graça. Olha-me com um certo espanto. Não sabe o que falar... Não entendo, no momento, sua reação e vou embora da loja um tanto pensativo.

Depois, em casa, relembrando a cena, me dou conta do mal-entendido criado por um conjunto de jargões mal-empregados:

— Eu de roupa branca.
— Meu bip é acionado.
— Peço para ligar com certo ar de urgência.
— Pergunto se "*é da sala da operação?*"
— Torno a perguntar se "*abortou ou foi morto?*"
— Digo com toda a tranqüilidade que "*Se abortou não tem problema. Era só um teste. Podemos fazer outro amanhã...*"

O que mais poderia parecer a alguém que presenciasse tal diálogo e não tivesse todas as informações sobre o assunto?

Com esse exemplo, pode-se perceber que a ambigüidade se estabelece a partir do instante em que duas, ou mais, interpretações são possíveis para um mesmo conjunto de elementos. No caso citado, perguntar "*se abortou ou foi morto*" poderia tanto ter uma conotação médica como técnica. Conhecendo-se os interlocutores e os diálogos completos ninguém teria dúvida sobre o contexto e sobre o significado da frase, mas conhecendo somente parte ...

Este é o ponto que devemos ter em mente ao construirmos nossos modelos :

> *A representação e denominação dos elementos que compõem um modelo de dados não são suficientes, na grande maioria dos casos, para traduzir todos os conceitos necessários, podendo vir a ser gerado um certo grau de ambigüidade em sua interpretação se cuidados adicionais não forem tomados.*

Para que se tenha um exemplo concreto dessa situação aplicada à modelagem de dados, vamos propor um ambiente contendo apenas duas entidades a serem modeladas. Ao observar esse ambiente e abstrair características dos objetos nele existentes, poderíamos ter chegado à seguinte situação:

- **Entidades identificadas:**

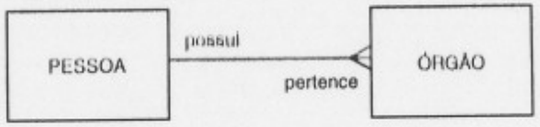
ÓRGÃO e PESSOA

- **Atributos identificados:**

Para ÓRGÃO:
— identificação do ÓRGÃO
— nome do ÓRGÃO
— grau de classificação do ÓRGÃO

Para PESSOA:
— identificação da PESSOA
— nome da PESSOA
— nível social
— telefone para contato

- **Modelo gerado:**



- **Interpretação do modelo:**

— uma PESSOA possui vários ÓRGÃOS *e*
— um ÓRGÃO pertence a uma PESSOA

Pode-se perceber que temos a correta identificação das entidades e seus atributos. Poderíamos ter, também, representado-as através de uma outra notação gráfica qualquer, mas, de qualquer modo, continuaríamos com certo grau de ambigüidade em nosso modelo enquanto não definíssemos através de um dicionário de dados o que significam ÓRGÃO e PESSOA.

ÓRGÃO pode ser, dependendo de informações adicionais que possamos acrescentar, quaisquer um destes conceitos:

1. Uma abstração de todos os sintetizadores e instrumentos musicais eletrônicos acionados via teclado e que são adquiridos por músicos em lojas especializadas.

2. Uma abstração de setores e repartições funcionais de uma empresa, os quais são comandados por funcionários de nível gerencial.

3. Uma abstração de todos os itens passíveis de serem doados por um ser humano após detectada sua morte cerebral.

4. E talvez outras abstrações particularmente diferentes mas passíveis de serem expressadas através do mesmo termo ÓRGÃO...

Assim, independentemente do nosso grau de certeza sobre os objetos abstraídos e representados em nosso modelo, devemos dar especial atenção à definição, ou descrição, do conceito que cada elemento do modelo procura expressar. Essa definição deve trazer informações adicionais que garantam a interpretação e compreensão inequívoca de um mesmo modelo tanto para a pessoa que o tenha criado, como para um leigo que não tenha tido qualquer contato anterior com o ambiente modelado.

Algumas regras podem ser estabelecidas baseadas nas considerações já feitas até o momento. Essas regras devem ser utilizadas durante o processo de modelagem visando a aumentar o grau de fidelidade na interpretação de nossos modelos:

1. Nenhum modelo é suficientemente claro se não for acompanhado de uma definição formal dos elementos (um dicionário de dados, ou similar).

2. Conceitos que possam parecer triviais a quem está modelando não serão do mesmo modo triviais a outras pessoas que não tenham o conhecimento prévio do assunto tratado. Não existem, em princípio, conceitos triviais a todas as pessoas. O que é trivial para uma pessoa poderá não ser a outra, portanto, a definição será necessária de qualquer modo. Assumir que um conceito é trivial e não precisa ser detalhado é assumir o risco de futuras interpretações diferentes daquelas que esperávamos. Não corra riscos!

3. O estabelecimento de definições completas e inequívocas é extremamente importante. Devemos buscar durante essa definição meios para sermos precisos, claros e específicos. Devemos procurar conceituar os elementos mostrando:

   - Regras nas quais eles se encaixam.
   - Exceções às regras, quando aplicáveis.
   - Exemplos ilustrativos.
   - Correlação entre conceitos.
   - E qualquer outro tipo de informação que contribua para a compreensão.

Exemplo:

Para caracterizar o objeto ÓRGÃO observado em um ambiente que pretendemos modelar, poderíamos utilizar, além da representação gráfica feita baseada em alguma técnica, uma definição do tipo:

ÓRGÃO

- **Atributos:**

Identificação do ÓRGÃO. Sigla formada pelas principais letras significativas do nome do ÓRGÃO que serve como identificador e diferenciador entre os diversos órgãos da empresa. Ex.: DIROP.

Nome do ÓRGÃO. Denominação, ou título, atribuído ao ÓRGÃO baseado em sua finalidade ou função principal dentro da empresa. Essa denominação já incorpora o nível hierárquico do próprio ÓRGÃO. Ex.: "Diretoria Operacional".

Grau de classificação do ÓRGÃO. Indicativo do grau de autonomia de um determinado ÓRGÃO com relação a investimentos e gastos em atividades internas. Esse grau é definido no instante da criação do ÓRGÃO e pode vir a ser alterado em função de determinações da presidência da empresa. Ex.: "R — restrito; I — irrestrito".

- **Definição:**

Representa todo e qualquer setor ou repartição funcional existente no organograma da empresa. Os setores informais, isto é, aqueles não criados por Ato de Serviço, mas que tenham representatividade funcional também serão considerados nesse enquadramento.

- **Exemplos:**

CPD	— Centro de Processamento de Dados	Grau R
DIRAD	— Diretoria Administrativa	Grau I
SDES	— Setor de Despacho de Mercadorias	Grau R

- **Correlação:**

Ver também ESTRUTURA FUNCIONAL.

Nesse exemplo, fica fácil caracterizar o que é, e o que não é, um ÓRGÃO. Pouca, ou nenhuma, chance para interpretações equivocadas poderá surgir. Mesmo um leigo sendo apresentado pela primeira vez ao modelo ÓRGÃO — PESSOA poderá ser capaz de interpretar de modo correto os conceitos apresentados.

Veremos, mais a frente, que também os atributos e relacionamentos existentes entre os objetos deverão ser conceituados e definidos formalmente. Com isso, poderemos assegurar uma melhor compreensão dos nossos modelos, tornando-os impessoais. As definições e elementos agregados para a correta conceituação dos objetos deverão seguir o seguinte princípio:

Toda informação útil deverá ser agregada, mas lembre-se: Nem toda informação é útil!

Isso, em outras palavras, significa que agregar informações sem cunho prático, simplesmente para cumprir a finalidade de "dicionarizar" um objeto, não traz grandes benefícios. O benefício será tanto maior quanto maior for a objetividade e utilidade das informações agregadas. Toda informação que contribua na conceituação e preservação de conhecimento sobre o objeto deverá ser alvo de atenção, pois poderá ser extremamente útil na interpretação do modelo. Definições do tipo "ÓRGÃO é o conjunto de órgãos da empresa", apesar de ser bastante freqüente, não acrescenta qualquer informação útil em termos conceituais, logo deverão ser evitadas.

Regras Básicas para a Dicionarização de Objetos (Entidades/Classes)

Uma das principais tendências quando passamos a dicionarizar um objeto é nos referenciarmos a ele como um meio de armazenamento e não como o próprio objeto que ele representa.

Assim, por exemplo, para um objeto ATLETA *não deveríamos* definir nosso dicionário como:

"Contém os dados relativos aos atletas que participam do torneio."

O que se deseja conhecer não é a finalidade do objeto (conter os dados de ...) mas sim:

- O que é um ATLETA?
- O que faz um ATLETA?
- Para que serve um ATLETA?
- O que engloba a categoria dos ATLETAS em nosso caso?
- O que exclui a categoria dos ATLETAS em nosso caso?
- Quando alguém passa a ser, ou deixa de ser, um ATLETA?

Assim, a dicionarização adequada deveria responder a essas perguntas, contendo um texto que fosse algo como:

É todo o funcionário da empresa que, para fins desportivos, representa-a em uma ou mais modalidades individuais ou coletivas. São considerados atletas todos aqueles que participam oficialmente do grêmio desportivo da empresa, quer por adesão voluntária ou por indicação de terceiros. Não é permitida a participação de pessoas externas à empresa como atletas.

Perceba que se também, agora, ler a definição de ATLETA, estará compreendendo o conceito envolvido e modelado sob o ponto de vista de uma dada empresa. Com certeza, em outros ambientes, existirão outras definições diferentes e válidas de ATLETA. É justamente isso que desejamos: estabelecer qual é o *nosso* conceito, em *nosso* mundo observado.

Requisitos a Serem Atendidos na Dicionarização

Para a dicionarização de um objeto, ou entidade, deveremos ter em mente algo que satisfaça, senão todas, a grande parte das seguintes perguntas:

- O que é o elemento?
- O que faz o elemento?
- Para que serve?
- O que engloba essa categoria de elementos?
- O que está excluído dessa categoria?
- Quando alguém passa a ser, ou deixa de ser, um elemento desse tipo?
- Sua permanência nessa categoria é imutável?

Considerações sobre Entidades Fortes e Entidades Fracas

Alguns autores adotam, para fins de caracterização de uma entidade, um critério que as classifica em *fortes* ou *fracas*. Esta caracterização se dá através da análise de existência de duas condições básicas:

- Dependência de existência.
- Dependência de identificador.

Dizemos que uma entidade é fraca se um desses dois tipos de dependência se verificar entre uma entidade A e uma entidade B. Normalmente, esse enquadramento está associado ao ambiente de projeto lógico, apesar de também poder ser aplicado ao projeto conceitual.

Dependência de Existência

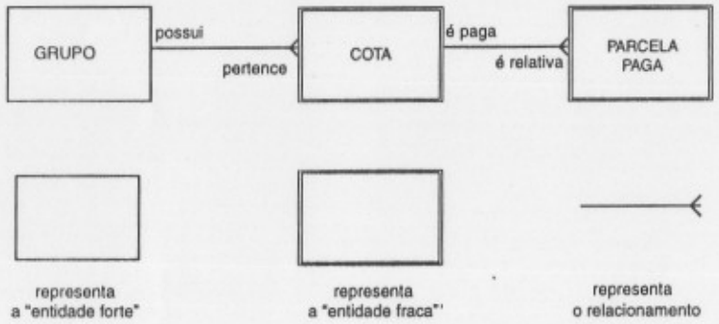
Essa situação denota o estabelecimento de um vínculo de existência entre dois tipos de entidades. Por exemplo, se B depender de A para existir, teremos em B uma entidade fraca, enquanto que A, se não depender de ninguém para existir, será considerada uma entidade forte.

Imagine o caso em que estejamos modelando um ambiente de uma administradora de consórcios. Nesse ambiente, detectamos a existência dos seguintes elementos: GRUPO, COTA e PARCELA PAGA. Sabemos que

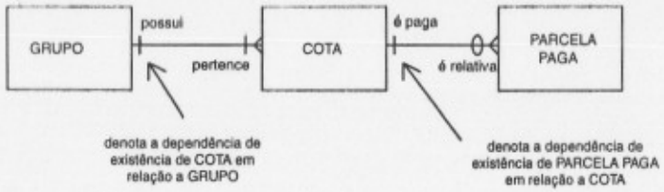
um GRUPO é formado por COTAS e que uma COTA, desde que vendida, passa a ter um histórico de pagamentos efetuados mês a mês no qual constam o valor, a data, o agente arrecadador entre outras informações. Vejamos agora o que podemos caracterizar com relação a entidades fortes e entidades fracas.

Uma COTA só existe desde que seu GRUPO tenha sido criado. Se acabarmos com o GRUPO, as COTAS não poderão continuar a existir sem a presença desse GRUPO. Assim também as PARCELAS PAGAS de cada uma das COTAS só pode existir desde que a COTA previamente exista. Se em um GRUPO não tivermos a COTA número 50, então não poderemos ter pagamentos sendo feitos para essa COTA. Perceba que nesses dois casos temos uma situação de dependência de existência.

Para denotar essa situação, pode-se utilizar o critério de entidades fortes e entidades fracas. Graficamente teríamos algo como:



Outro critério, que nos parece mais adequado, para denotar essa mesma situação é a definição de maiores detalhes para a associação existente entre os elementos. Veremos, quando tratarmos os relacionamentos, que poderemos explicitar os *graus máximos e mínimos de participação no relacionamento*. Os detalhes a serem fornecidos para um dado relacionamento farão com que seja possível assimilar os mesmos conceitos traduzidos pela idéia de "entidade forte" e "entidade fraca" sem que seja necessário caracterizá-las de antemão.



Como essa caracterização só se dá em função da existência de um relacionamento entre duas entidades (GRUPO *versus* COTA e COTA *versus* PARCELA PAGA) e como, eventualmente, alguns desses relacionamentos podem vir a deixar de existir em nosso ambiente observado, entendemos que afirmar que uma entidade é forte ou fraca é, no mínimo, classificá-la em função de um critério alheio a sua própria existência.

Se o relacionamento é quem define a característica forte ou fraca para uma entidade, por que não deixar então que ele mesmo (o relacionamento) explicite essa condição durante seu mapeamento? Por que fazer com que a entidade tenha que ser modelada de diferentes modos se o relacionamento é quem define sua característica?

Não estamos com isso afirmando que esse tipo de representação, ou critério de modelagem esteja incorreto, mas sim que, conceitualmente, ele pode ser melhorado. Alguns autores, durante o processo de modelagem, abordam fielmente o conceito de *entidades fortes* e *entidades fracas*. O próprio Peter Chen, no trabalho que deu origem a todo o processo de abordagem entidade-relacionamento, dá ênfase a esse ponto.

Nosso entendimento é que, inicialmente, essa abordagem justificava-se pela inexistência de outros recursos para representação de uma realidade desse gênero. Hoje, como já dispomos de recursos para que os relacionamentos denotem esse fato, podemos fazer com que a abordagem sugerida pelo próprio Peter Chen seja otimizada. Muitas vezes, aqueles que têm sido os maiores defensores da manutenção do conceito de entidade forte e entidade fraca utilizam, também, os recursos de graus máximos e mínimos de relacionamentos não percebendo que, com isso, tornam desnecessária a definição dos conceitos forte e fraca.

Dependência de Identificador

Esse segundo critério de definição de entidades fortes e entidades fracas, assim como o primeiro, também pode, sob o ponto de vista conceitual, ser considerado dispensável. Ele tem sua importância reconhecida sob o ponto de vista de projeto lógico, onde as chaves identificadoras, utilizadas como diferenciadores entre instâncias dos elementos, ou como método de endereçamento de registros, passam a ter papel vital durante o processo de projeto de estruturas de dados.

Entretanto, para fins de modelagem conceitual e, por conseguinte, para o correto nivelamento das atividades executadas em cada um dos instantes (projeto conceitual, projeto lógico e projeto físico), devemos desconsiderar esse tipo de influência para nosso modelo. Seria extremamente ineficaz inserirmos elementos diferenciados em nossos modelos (entidades fortes e fracas) simplesmente baseados nos conceitos de identificadores das entidades. Estaríamos precipitando as atividades de projeto lógico e assim fugindo de nosso principal foco de atenção, que deve ser *entender e modelar o ambiente observado.*

Somente para que não passemos em branco por es. ıtem, iremos elaborar um exemplo desse tipo de dependência para que se possa perceber sua aplicabilidade. Vamos imaginar que tenhamos recebido a incumbência de modelar um ambiente onde DECLARAÇÃO DE IMPOSTO DE RENDA e CONTRIBUINTE sejam nossos principais objetos observados. Poderíamos estar falando de um escritório contábil, de uma empresa especializada em consultoria contábil ou da própria Receita Federal.

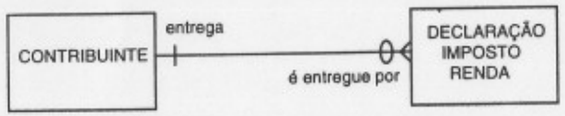
Em nosso levantamento poderíamos ter identificado que para o CONTRIBUINTE teríamos dados relevantes a serem considerados tais como: CPF, nome, endereço, telefone, etc. Para a DECLARAÇÃO DE IMPOSTO DE RENDA teríamos algo como: ano-exercício, ano-base, valor, indicação de restituição ou pagamento, número de cotas e data da entrega, etc. Sob o ponto de vista de análise de entidades fortes e entidades fracas, especialmente no tocante à dependência de identificadores pode- ríamos ter definido que:

CONTRIBUINTE é identificado por:

CPF

DECLARAÇÃO DE IMPOSTO DE RENDA é identificada por:

CPF do contribuinte
ANO de exercício



Perceba que a DECLARAÇÃO por si só não tem, nesse caso, um conjunto de identificadores próprios. Ela depende do identificador do CONTRIBUINTE para a formação de sua chave de identificação. Isso caracteriza, portanto, uma *dependência de identificação,* tornando, assim, a entidade DECLARAÇÃO uma entidade fraca. Entretanto, o que aconteceria se tivéssemos definido que a identificação das DECLARAÇÕES seria feita pela utilização de um número seqüencial apropriado no momento de sua recepção na agência bancária? Nesse caso, já não teríamos mais uma dependência de identificador e a entidade DECLARAÇÃO seria, portanto, uma entidade forte sob o ponto de vista desse critério. Isso demonstra ser esse critério bastante subjetivo e dependente de formas de implementações planejadas para as estruturas de dados.

Por isso, as influências desse tipo de análise em nosso modelo deveriam ser evitadas na fase de modelagem conceitual. Suas características não retratam o universo observado e sim um modo específico de tentar implementá-lo através de identificadores. Atualmente, sob o ponto de vista de

modelagem orientada a objetos, esse critério ficou ainda mais subjetivo, pois o tratamento dado aos conceitos de identificadores evoluiu e aproximou-se bastante do que propomos como modelo conceitual. Isso significa que detalhes relativos a chaves identificadoras são de muito baixo nível para que nos preocupemos com sua existência neste instante.

Lembre-se de que:

> *Um universo qualquer que esteja sendo observado não vive em função dos identificadores de seus elementos, mas sim da existência desses elementos e de seus relacionamentos.*

Portanto:

> *Esqueça, neste instante, toda e qualquer preocupação com chaves identificadoras! Isso lhe trará preocupações suficientes quando você estiver envolvido com o projeto de suas estruturas de dados! Não sofra por antecipação!*

Atributos

O Papel dos Atributos

Ao observarmos objetos em um ambiente, estaremos, na verdade, reconhecendo tais elementos através da identificação de suas características próprias. Essas características, inerentes a cada um desses objetos, serão, em princípio, comuns a todos os objetos, ou elementos individualizados, pertencentes a um mesmo conjunto.

Assim, ao observarmos um objeto que passa pela rua a nossa frente, estaremos sendo capazes de, já num primeiro instante, definir que o objeto visto é um CARRO e não uma PESSOA. Esse tipo de reconhecimento é realizado a partir da verificação de uma série de características para as quais atentaremos e julgaremos.

Características que nos definem um CARRO serão particulares e aplicáveis à quase totalidade dos carros. Também o mesmo será verdadeiro com relação a PESSOAS. Poderemos nos defrontar com casos onde existam, eventualmente, exceções de existência de uma característica comum a um conjunto para algum dos elementos desse mesmo conjunto.

Se tivéssemos como características de CARRO e PESSOA os atributos a seguir relacionados, poderíamos reconhecer entre um e outro tipo de objeto facilmente. O objeto que apresentasse marca, placa, modelo e ano

de fabricação como atributos nunca seria uma PESSOA. Já um objeto com cor de cabelo, comprimento dos cabelos, estatura, peso e idade, por sua vez, nunca seria um CARRO.

CARRO	PESSOA
marca: Chevrolet	cor dos cabelos: preta
placa: ADO-0911	comprimento dos cabelos: curto
modelo: Kadett GL	estatura: 1,77 m
ano de fabricação: 1996	peso: 72 Kg

Como exceções às características apresentadas pelos objetos CARRO e PESSOA, poderíamos citar:

- Um carro militar que não possui a característica "placa".
- Uma pessoa careca que, portanto, não apresenta a característica "cor dos cabelos".

Através da análise dos atributos que um objeto detém, podemos enquadrá-lo como pertencente a um, ou outro, conjunto distinto. Entretanto, a simples observação de uma série de atributos pode não ser suficiente para determinar seu enquadramento. Muitas vezes os atributos observados podem ser comuns a mais de um tipo de objeto. Isso pode ser verdadeiro em duas situações:

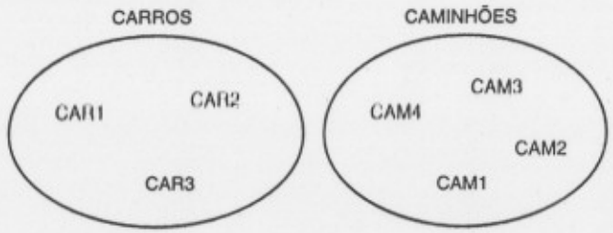
1. O universo de atributos definidos como caracterizadores de um tipo de objeto é muito restrito ou insuficiente para determiná-lo totalmente.
2. O nível de abstração utilizado para observação e caracterização dos objetos não está corretamente nivelado com a nomenclatura associada aos conjuntos.

Exemplo:

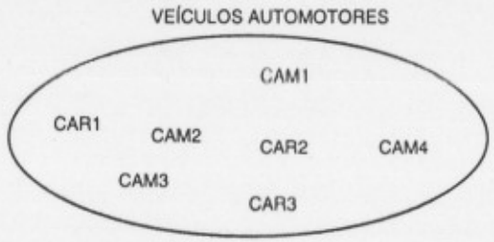
Um CAMINHÃO ao ser observado e caracterizado pelos atributos marca, placa, modelo e ano de fabricação será facilmente distinguido de uma PESSOA, mas acabará por ter o mesmo enquadramento dado a um CARRO. Entretanto, sabemos que qualquer que seja a instância de CAMINHÃO, esta não pertencerá ao conjunto de CARROS.

Nesse caso, existem duas alternativas possíveis para sanar esse tipo de deficiência:

1. Buscar outros atributos que nos façam distinguir objetos que pertençam ao conjunto de CAMINHÕES daqueles pertencentes ao conjunto de CARROS, criando, portanto, distintamente estes dois conjuntos:



2. Identificar, através de uma abstração de maior nível, outro conjunto ao qual os objetos do tipo CAMINHÃO e do tipo CARRO possam pertencer. Neste exemplo, o conjunto generalizado poderia ser denominado VEÍCULOS AUTOMOTORES. A esse conjunto poderiam pertencer, então, todos os objetos do tipo CARRO e do tipo CAMINHÃO, mesmo que, em um primeiro instante, não tivéssemos preocupação em diferenciá-los.

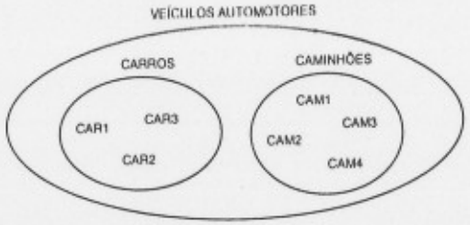


Entre essas duas alternativas, a que parece seguir melhor a linha de percepção natural dos objetos é a primeira. Por quê? Porque se no próprio enquadramento de objetos aos conjuntos já estamos sendo capazes de afirmar que temos CAMINHÕES e CARROS é porque, de alguma maneira, pela análise de seus atributos, estamos sendo capazes de diferenciá-los. Desse modo, bastará que enumeremos esses atributos utilizados na diferenciação e que os aloquemos no conjunto devido.

Veremos mais a frente, quando estivermos tratando de extensões do modelo de entidade-relacionamento, que o recurso de generalização e especialização de objetos pode nos levar a uma terceira alternativa que é:

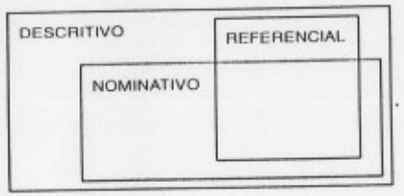
3. Definir um conjunto que mantenha as características, ou atributos, comuns aos objetos pertencentes a cada um dos subconjuntos e, também, criar os subconjuntos para que neles mantenhamos as

características específicas de cada um dos objetos. Nesse caso, teríamos o conjunto VEÍCULOS AUTOMOTORES com atributos aplicáveis tanto a CARROS como CAMINHÕES, e também os subconjuntos CARROS e CAMINHÕES, cada qual com seus atributos específicos.



Classificação dos Atributos Quanto a Sua Finalidade

Após termos identificado os atributos de cada um dos objetos, podemos, sob o ponto de vista de classificação quanto a sua finalidade, enquadrá-los em três grandes grupos:



A definição formal do grupo ao qual pertence cada um dos atributos identificados no modelo não é necessária para fins semânticos. O entendimento de um modelo se processará com ou sem esse tipo de informação. Se considerarmos, entretanto, a futura fase de derivação lógica, ou seja, aquela onde nosso interesse será a geração de estruturas de dados para implementação, a classificação prévia dos atributos nominativos e referenciais poderá contribuir no estabelecimento de chaves e chaves estrangeiras.

Como no instante em que se está processando esse tipo de classificação nosso interesse ainda é *puramente conceitual*, esse tipo de formalismo poderá ser considerado apenas como simples elemento de apoio ao entendimento da característica principal de um atributo.

Atributos Descritivos

Os atributos classificados como descritivos são aqueles que, sob o enfoque funcional, aplicam-se à representação das características intrínsecas dos objetos. Todo e qualquer atributo que seja capaz de demonstrar, ou

representar, características formadoras, ou pertencentes, a um objeto poderá vir a ser enquadrado como descritivo.

Partindo-se dessa definição, em princípio, todos os atributos podem ser classificados como descritivos. Isso é verdadeiro e completamente coerente com a própria definição de atributo. O que fará com que alguns atributos deixem de ser classificados como descritivos e passem a pertencer aos demais grupos será a existência de características funcionais adicionais que poderão ser captadas através de suas interpretações.

Exemplo:

Atributo Descritivo	Exemplo	Objeto
data de nascimento	25-MAR-62	PESSOA
quantidade de cilindros	6	VEÍCULO AUTOMOTOR
índice de reajuste	10,33	CONTRATO
raça	FILA	ANIMAL

Como procuramos deixar visível através dos exemplos, um atributo descritivo não precisa ser, sob o ponto de vista de característica física do dado a ser armazenado, do tipo string, literal, caracter, char, longchar, x(...), ou qualquer outro tipo associado a um conjunto de caracteres. Apesar de o termo *descritivo* induzir grande parte das pessoas a pensar em "um texto que descreve", esta não é a idéia correta a ser estabelecida.

Um atributo descritivo pode ser uma data, um número inteiro, um número fracionário, uma cor, uma raça, ou qualquer outro tipo de domínio discreto ou contínuo. A idéia que deve ser associada ao termo *descritivo* é, na verdade, *aquilo que é capaz de demonstrar características de um objeto*.

Atributos Nominativos

O grupo dos atributos nominativos engloba todos aqueles atributos que, além de cumprirem a função de descritivos, também servem como definidores de nomes ou rótulos de identificação aos objetos aos quais pertencem. Logo, qualquer atributo que possa de certa maneira identificar um objeto, mesmo que não de maneira unívoca, é dito um atributo nominativo.

Sob esse enfoque, os fortes candidatos a atributos nominativos serão sempre aqueles que, já no nível conceitual, são reconhecidos e denominados como *nome do ...*, *código do ...*, *número do ...*, *identificador do ...*, *sigla do ...*, entre outros.

Durante o processo de projeto lógico de bancos de dados relacionais, esses atributos poderão ser analisados como candidatos a chaves candidatas.

Alguns deles poderão vir a ser descartados como chaves candidatas, pois não atenderão ao critério de geração de chaves unívocas.

Exemplo:

Atributo Nominativo	Exemplo	Objeto
número da carteira de trabalho	867301	FUNCIONÁRIO
nome do funcionário	JOÃO DA SILVA	FUNCIONÁRIO
número de série	110.764-E	EQUIPAMENTO
sigla do ÓRGÃO	CPD	ÓRGÃO FUNCIONAL
prefixo do avião	PT-EGC	AERONAVE

Também nesse tipo de classificação é freqüente haver algum tipo de interpretação indevida. Os dois pontos que mais freqüentemente geram dúvidas são:

1. Muitas pessoas são induzidas a pensar que atributos nominativos exigem a presença de "nomes", ou seja, de um "literal que exprima um nome". Isso não é verdadeiro, pois, apesar de "nomes" se enquadrarem realmente como atributos nominativos, também atributos que se apresentem como códigos numéricos ou alfanuméricos são válidos. Basta que consigamos estabelecer um atributo como definidor de um "rótulo" para um dado objeto para que ele possa ser enquadrado como nominativo.

2. Outra dúvida que surge, muitas vezes, é com relação à necessidade dos atributos nominativos serem atributos que identifiquem de maneira unívoca as instâncias dos objetos aos quais pertencem. Insistimos em que não há obrigatoriedade nesse sentido. Se o atributo "nome do funcionário" é capaz de rotular uma instância de um objeto do tipo FUNCIONÁRIO, então, mesmo que duas instâncias venham a receber o mesmo rótulo (no caso de homônimos), teremos cumprido nosso objetivo através desse atributo. O conceito de identificadores unívocos é uma exigência somente do nível de projeto lógico, e assim mesmo, quando associado a alguns tipos de tecnologias, como a de bancos de dados relacionais. Caso estivéssemos, por exemplo, visando a um ambiente de implementação em um banco de dados orientado a objetos, não teríamos, nem no nível de projeto lógico, esse tipo de restrição, o que demonstra a total coerência em classificarmos atributos que não definam chaves unívocas como atributos nominativos.

Atributos Referenciais

Os atributos classificados como referenciais são aqueles que não pertencem propriamente ao objeto onde estão alocados, mas fazem algum tipo de citação, ou ligação desse objeto com um outro objeto. Pode-se dizer que não estão, na verdade, definindo qualquer característica intrínseca do objeto mas sim explicitando um relacionamento existente entre objetos.

Exemplo:

Atributo Referencial	Exemplo	Objeto no Qual Está Definido	Objeto de Origem
Local de trabalho	Fábrica 1	FUNCIONÁRIO	FÁBRICA
Nome do fabricante	Fiat	CARRO	FABRICANTE
Nome do proprietário	José de Souza	IMÓVEL	PESSOA
Cidade onde nasceu	Curitiba	FUNCIONÁRIO	CIDADE

Ao identificarmos um atributo referencial estaremos assumindo a existência de um outro objeto ao qual nos referenciamos. Esse outro objeto, o objeto de 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. Em algumas situações estaremos percebendo a existência de atributos referenciais e para cada uma delas algumas estratégias de tratamento são possíveis:

1. Atributos referenciais no modelo conceitual: Na fase de modelagem conceitual, a existência de atributos referenciais pode demonstrar a não normalização dos dados, ou seja, a existência de dados alocados em lugares não adequados sob o ponto de vista de manipulação de dados. Como o modelo conceitual não precisa necessariamente estar normalizado, esta poderá ser uma situação com a qual conviveremos conscientemente. Entretanto, não devemos deixar de identificar esses atributos como referenciais justamente para podermos, quando necessário, buscar a normalização do modelo. Essa situação será verdadeira, desde que não tenhamos criado em nosso modelo o objeto origem desses atributos, pois caso ele exista, então nosso modelo conceitual não deverá conter atributos referenciais a esse objeto, devendo ser procedidas a remoção e alocação desses atributos em seu próprio objeto e a criação de um relacionamento entre eles.

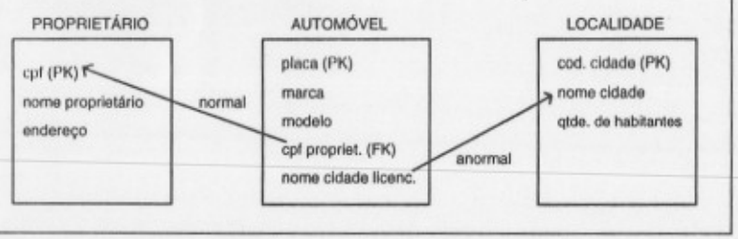
Exemplo:

ENTIDADE: AUTOMÓVEL

Atributo	Exemplo	Classificação
Placa	ACB-3431	Nominativo
Marca	Kadott	Descritivo
Modelo	SL	Descritivo
Chassi	9BK0043029MJ003	Nominativo
Nome do proprietário	Paulo	Referencial
Nome do fabricante	Chevrolet	Referencial

2. Atributos referenciais no modelo lógico: A existência de atributos referenciais no modelo lógico, quando voltado ao ambiente relacional, poderá ser normal desde que esses atributos caracterizem-se como chaves estrangeiras. Outros atributos referenciais que não sejam chaves estrangeiras estarão denotando falta de normalização e, portanto, deverão ser tratados sob esse ponto de vista.

Exemplo:



Dicionarização dos Atributos

A dicionarização dos atributos identificados nos objetos por nós observados é, complementarmente à dicionarização dos próprios objetos e entidades, o único modo efetivo de caracterização do universo observado.

Freqüentemente, quando se trata a dicionarização dos atributos acaba-se por cometer um tipo de falha que compromete bastante o resultado final de nosso modelo: a simplificação. Essa simplificação advém do fato de que o indivíduo envolvido com a concepção do modelo, normalmente, acaba por envolver-se suficientemente com o ambiente que está modelando e assim passa a enxergá-lo de modo muito claro. Isso seria excelente, sob o

ponto de vista de m lagem, a não ser pelo fato de que suas definições de atributos passam a ser, desse momento em diante, extremamente breves, pois para ele, isso lhe basta.

Entretanto, devemos lembrar que não estamos dicionarizando, somente, para nosso próprio uso e benefício. Devemos pensar em todos aqueles que não tiveram a chance de envolver-se com o ambiente e conhecer, de perto, suas características e detalhes. Assim, nossa dicionarização deve procurar trazer ao conhecimento público toda e qualquer informação que seja de valia para o processo de compreensão e unificação de conceitos.

O que temos visto, muito freqüentemente, é que os dicionários de dados, principalmente de atributos, acabam por se transformar em "tradutores de mnemônicos ou data-names". Um caso muito comum pode ser exemplificado, utilizando-se a entidade ATLETA já descrita anteriormente em outro exemplo. Imaginemos que para essa entidade temos um atributo identificado que é:

Atributo: IND-TIPO-ATLETA.

Uma dicionarização extremamente simplificada, portanto, *não recomendada*, seria:

Definição: *"É o indicador do tipo de atleta existente na empresa."*

Perceba que algumas perguntas não conseguem ser respondidas com essa definição. Por exemplo:

- Qual é a finalidade em diferenciar os atletas em tipos?
- Como são enquadrados os atletas em tipos?
- O que significa um tipo de atleta?
- Quais as diferenças entre os diversos tipos?
- Quem define que tipo será alocado a um atleta?
- Quando é atribuído esse tipo ao atleta?
- Depois de definido um tipo para o atleta, esse tipo pode mudar?
- Qual o papel do tipo de atleta no ambiente? E no sistema?

Assim, a dicionarização do atributo IND-TIPO-ATLETA *deveria ser* algo como:

Define, dentre o grupo geral dos atletas da empresa, quais são aqueles que participam como titulares ou como reservas em suas respectivas equipes. Todo atleta é, inicialmente, cadastrado sem um enquadramento específico e, após análise, é reenquadrado pela comissão técnica como titular ou reserva. Essa diferenciação visa a orientar o processo de escalação das equipes para eventos externos

à empresa, pois somente os titulares têm suas horas de ausência abonadas.

Além disso, deveríamos complementar a dicionarização, quando aplicável, definindo o domínio discreto que o atributo pode conter. Em nosso exemplo poderíamos ter:

Domínios Válidos: T — Titular
R — Reserva
N — Não enquadrado

Assim, de posse de todas essas informações, qualquer indivíduo, mesmo não conhecendo o ambiente de controle de atividades desportivas da empresa X, poderia ter uma clara visão do que representa esse atributo no modelo.

Relacionamentos

Identificação dos Relacionamentos

Até o momento, abordamos a identificação e o mapeamento dos objetos percebidos em um universo observado. Vimos que devemos identificá-los, caracterizá-los e representá-los. Cada objeto é percebido por suas características próprias e assim, então, mapeado.

Outro ponto a ser mapeado, já citado na definição da "Lei do Mundo", é o *relacionamento* estabelecido entre os objetos. Esses relacionamentos existem, também, por características intrínsecas de cada um dos objetos, ou do meio onde estão situados. Ao observarmos os objetos e reconhecê-los estaremos, quase que imediatamente, reconhecendo as relações existentes entre eles. Muitas vezes a própria observação de um relacionamento será o ponto de partida para a identificação dos objetos que dele participam.

Através do mapeamento dos relacionamentos estaremos sendo capazes de demonstrar como um objeto se comporta em relação aos demais, qual seu grau de dependência de outros objetos, qual a associação de dados existente entre eles, entre outros fatores. Devemos estar atentos para que o mapeamento desses relacionamentos possa agregar todas as características semânticas existentes em um certo tipo de associação.

Cada objeto poderá se relacionar com diversos outros objetos independentemente de seu tipo. O que estabelecerá associações válidas ou não será simplesmente o grau de fidelidade e completeza que consigamos atingir durante o processo de modelagem. Isso significa que, se um relacionamento mapeado espelha a realidade, ele poderá envolver quaisquer tipos ou instâncias de objetos. Para fins de classificação podemos eleger dois grupos básicos de relacionamentos:

1. Relacionamentos entre instâncias de objetos de diferentes tipos.
2. Relacionamentos entre instâncias de um mesmo tipo de objeto.

Relacionamento entre Objetos de Diferentes Tipos

Esse tipo de relacionamento será o mais freqüentemente observado. Nesse tipo de relacionamento, estaremos associando instâncias de um objeto de um tipo a outras de outro tipo. Isso pode ser facilmente compreendido se, para fins de exemplo, estabelecermos um relacionamento entre PESSOA e CARRO. A esse relacionamento atribuiremos a finalidade de expressar o conceito de *propriedade*.

Assim, se desejarmos ter, conceitualmente, representado um ambiente observado onde "*Eduardo é proprietário de um jipe amarelo*", poderemos nos valer da seguinte estratégia:

1. Identificar os objetos envolvidos

PESSOA, instanciada por "Eduardo"
CARRO, instanciado por "jipe amarelo"

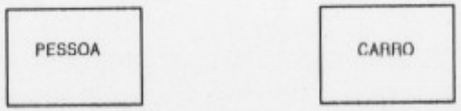
em que temos para cada tipo de objeto (PESSOA e CARRO) uma instância específica (Eduardo e jipe amarelo).

2. Caracterizar os objetos

PESSOA, caracterizada por:
nome, data de nascimento, número CPF

CARRO, caracterizado por:
marca, cor, ano de fabricação, modelo

3. Representar os objetos



4. Identificar o relacionamento entre os objetos

Relacionamento: PESSOA *é proprietária de* CARRO

5. Caracterizar o relacionamento entre os objetos

Nem toda PESSOA é proprietária de um CARRO
Um CARRO pode pertencer a uma PESSOA, ou não
Algumas PESSOAS têm mais de um CARRO
Se um CARRO pertence a uma PESSOA ele não pertencerá a mais ninguém

6. Representar o relacionamento

Nesse exemplo, procuramos mostrar o processo de identificação e caracterização de um relacionamento entre os objetos do tipo PESSOA e CARRO. Esse processo pode ser utilizado para mapeamento de qualquer relacionamento entre dois, ou mais, tipos de objetos e, também, para mapeamento de mais de um tipo de relacionamento entre os mesmos objetos. Assim, caso necessitássemos expandir nosso modelo representando também as observações

- Um carro é de propriedade de uma pessoa mas pode ser utilizado por diversas outras para locomoção.
- Uma pessoa utiliza um imóvel para morar.

teríamos que repetir os processos de identificação dos objetos participantes de cada um dos relacionamentos, de identificação das características, dos relacionamentos e de representação. Ao final, acabaríamos por obter algo como:

1. Identificar os objetos envolvidos

PESSOA
CARRO
IMÓVEL.

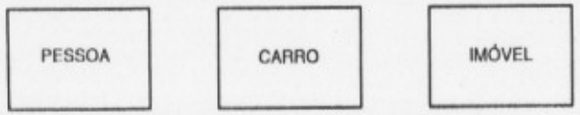
2. Caracterizar os objetos

PESSOA, já caracterizada anteriormente por:
nome, data de nascimento, número CPF

CARRO, já caracterizado anteriormente por:
marca, cor, ano de fabricação, modelo

IMÓVEL, caracterizado por:
localização, metragem, tipo de edificação, número de registro

3. Representar os objetos



4. Identific s novos relacionamentos entre os objetos

Relacionamento: PESSOA *utiliza* CARRO
PESSOA *utiliza* IMÓVEL

5. Caracterizar os relacionamentos entre os objetos

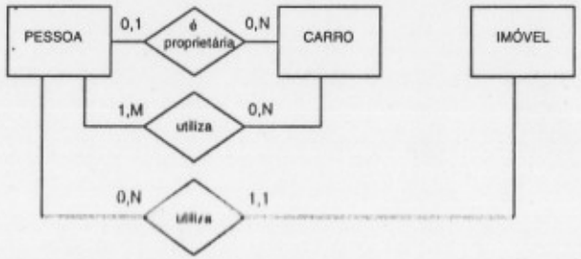
- PESSOA *utiliza* CARRO

Nem toda PESSOA utiliza um CARRO
Um CARRO pode ser utilizado por uma, ou mais PESSOAS
Algumas PESSOAS utilizam mais de um CARRO
Um CARRO sempre será utilizado por, pelo menos, uma PESSOA

- PESSOA *utiliza* IMÓVEL

Toda PESSOA utiliza um, e só um IMÓVEL para morar
Um IMÓVEL pode ser utilizado por uma, ou mais PESSOAS
Um IMÓVEL nem sempre será utilizado por uma PESSOA

6. Representar os novos relacionamentos no modelo já existente



Pode-se perceber por essa extensão em nosso modelo inicial que:

1. Entre dois tipos de objetos podem ser estabelecidos mais de um tipo de relacionamento, tal como "PESSOA *é proprietária de* CARRO" e "PESSOA *utiliza* CARRO".
2. Um objeto pode participar de diversos relacionamentos, cada um deles com diferentes tipos de objetos, tal como "PESSOA utiliza CARRO" e "PESSOA utiliza IMÓVEL".
3. Relacionamentos estabelecidos entre diferentes tipos de objetos podem expressar conceitos iguais ou semelhantes, ou eventualmente somente possuir o mesmo nome significando

conceitos ligeiramente diferentes, tal como "PESSOA *utiliza* CARRO" e "PESSOA *utiliza* IMÓVEL".

Relacionamento entre Objetos do Mesmo Tipo

Esse tipo de relacionamento caracteriza um caso especial onde no papel de objetos formadores, ou participantes, do relacionamento encontraremos o mesmo tipo de objeto. Esse tipo de associação de objetos também é denominado de *auto-relacionamento* ou *relacionamento recursivo*.

Apesar de se tratar de um caso especial, esse tipo de relacionamento pode ter o mesmo processo de abordagem para seu reconhecimento, caracterização e mapeamento. Assim, também a seqüência descrita anteriormente é valida:

1. Identificar o objeto envolvido no relacionamento.
2. Caracterizar o objeto.
3. Representar o objeto.
4. Identificar o relacionamento entre as instâncias do objeto.
5. Caracterizar o relacionamento.
6. Representar o relacionamento.

Pode se perceber que alguns pontos têm uma abordagem levemente diferenciada em relação ao processo de tratamento dos relacionamentos convencionais (entre dois diferentes tipos de objetos). O primeiro ponto diz respeito à existência de somente um objeto. Isso faz com que tenhamos que reconhecer diferentes papéis de um mesmo objeto no relacionamento. Depois, temos que caracterizar o objeto sob o ponto de vista de duplo participante na relação. Na seqüência temos de representar de algum modo o relacionamento estabelecido entre esse objeto e ele mesmo. Veremos como tratar cada um desses casos através de um exemplo.

Exemplo:

Suponhamos que tenhamos observado um ambiente de uma organização prestadora de serviços de vigilância. Nesse ambiente fomos capazes de reconhecer, dentre os vários objetos, um que denominamos de VIGILANTE. Percebemos também que dentre os vários relacionamentos estabelecidos entre o objeto VIGILANTE e os demais objetos existe um que determina a relação: "*um vigilante é substituído por outro vigilante*". Isso ocorre em algumas situações, tais como ao final do turno, na falta ao trabalho, no afastamento por doença, etc.

Podemos perceber que "*VIGILANTE é substituído por VIGILANTE*" é um relacionamento que se estabelece entre duas instâncias de objetos

de um mesmo tipo. Estamos associando, nesse caso, um vigilante a outro através de um relacionamento.

Seguindo os passos já definidos para o mapeamento do relacionamento teremos:

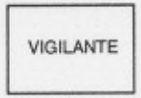
1. Identificar o objeto envolvido no relacionamento

VIGILANTE

2. Caracterizar o objeto

VIGILANTE caracterizado por:
identificação, nome, idade, especialização, disponibilidade, turno

3. Representar o objeto



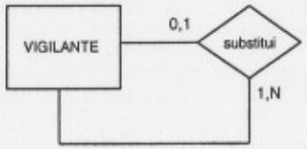
4. Identificar o relacionamento entre as instâncias do objeto

Relacionamento: VIGILANTE *substitui* VIGILANTE

5. Caracterizar o relacionamento

Um vigilante pode, ou não, ser substituído por outro
Um vigilante, se substituído, será somente por um outro
Um vigilante pode substituir um ou mais de seus colegas
Alguns vigilantes não substituirão ninguém em algumas situações

6. Representar o relacionamento



O mais importante no processo de modelagem dos relacionamentos é procurar manter o foco de atenção sobre os objetos observados e sobre o modo como eles se integram. Se você tiver a habilidade de perceber e distinguir os objetos em um ambiente, então o reconhecimento dos relacionamentos não será assim tão difícil.

Agora, pare e busque neste instante seu universo de observação. Pode ser sua sala de estudos, seu escritório, uma biblioteca onde você esteja, o saguão do aeroporto, o ônibus que o leva, neste instante, ao trabalho. Em cada um desses ambientes perceba os objetos nele contidos. Abstraia características. Enquadre esses objetos em conjuntos. Identifique o relacionamento entre eles.

Você irá perceber que os mais diversos objetos podem ser mapeados, tudo dependendo do ponto de observação estabelecido, dos requisitos, e de outras tantas características que já abordamos no processo de modelagem. Dentre os relacionamentos e objetos identificados você pode ter encontrado, por exemplo:

- Avião conduz passageiro.
- Livro é armazenado na estante.
- Aluno participa de aula.
- Recepcionista atende a cliente.

Denominação dos Relacionamentos

No processo de reconhecimento e identificação dos relacionamentos existentes entre os objetos devemos buscar uma denominação que represente o conceito observado. Essa denominação será, do ponto de vista semântico e conceitual, extremamente importante para a correta construção do modelo. Caso venhamos a denominar de modo inadequado um relacionamento, correremos o risco de transmitir, através do modelo, conceitos totalmente diferentes daqueles observados.

Basicamente, um relacionamento é expresso através de uma construção verbal. O termo designando o relacionamento poderá estabelecer uma forma ativa ou passiva para a aplicação do verbo desejado. Assim se a entidade PESSOA tem um relacionamento com a entidade CARRO por uma pessoa estar sendo transportada através desse CARRO, poderíamos ter as seguintes denominações alternativas:

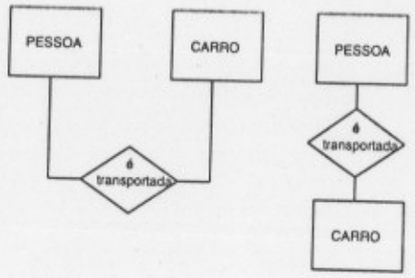
- Na forma passiva:
  PESSOA *é transportada por* CARRO

- Na forma ativa:
  CARRO *transporta* PESSOA

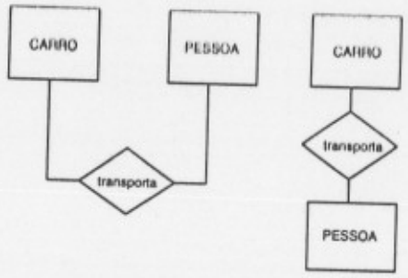
Essas duas denominações para um mesmo relacionamento podem ser utilizadas indistintamente, dependendo do ponto de vista que se queira abordar o relacionamento. Na verdade, alguns modelos, em função do próprio padrão de representação gráfica, utilizam os dois modos simultaneamente.

Mesmo que venhamos a utilizar somente uma das denominações em nosso modelo, a "leitura do modelo" poderá ser feita sob os dois aspectos. Apesar de não existir formalmente uma regra que assim estabeleça, normalmente, a denominação do relacionamento pode ser arbitrada em função do "sentido de leitura" do modelo. Pode-se considerar, para fins de facilidade de entendimento, que um relacionamento deva ser "lido" de cima para baixo ou da esquerda para a direita. Assim, se em nosso modelo o objeto PESSOA aparecer mais à esquerda ou mais acima do que o objeto CARRO, deveríamos denominar o relacionamento como "PESSOA é transportada por CARRO", já se CARRO aparecesse mais à esquerda ou mais acima do que PESSOA deveríamos denominar o relacionamento como "CARRO transporta PESSOA".

- **Denominação do relacionamento com PESSOA prevalecendo no papel de determinador do tempo verbal (mais à esquerda ou acima):**

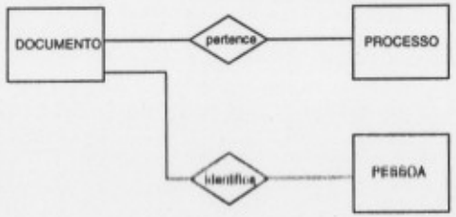


- **Denominação do relacionamento com CARRO prevalecendo no papel de determinador do tempo verbal (mais à esquerda ou acima):**



Nesse tipo de representação, temos como desvantagem a necessidade de estabelecer um sentido de leitura que defina o correto entendimento da realidade desejada. Isso parece fácil, mas nem sempre o é. Imagine que um

certo relacionamento descreve os conceitos "um documento pertence a um processo" e "um documento identifica uma pessoa". Esse modelo poderia ser representado seguindo o padrão preestabelecido da esquerda para a direita e de cima para baixo.

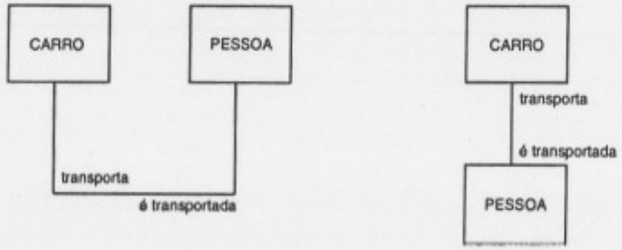


Agora imagine que durante o passar do tempo esse modelo foi sendo alterado, pela inclusão de novos conceitos, pela criação de novos relacionamentos e por outras atualizações que fizeram com que a disposição final das entidades em questão ficasse com novo aspecto:

Pode-se notar que pela movimentação das entidades durante o processo de realocação de espaços do desenho, acabamos por deixar, segundo as regras de leitura preestabelecidas, algumas anomalias em nosso modelo. Alguém que, sem conhecer outro tipo de referência, interprete o modelo acabará por assumir que *um processo pertence a um documento* e que *uma pessoa identifica um documento*, quando na verdade sabemos que o verdadeiro seria *um documento pertence a um processo* e *um documento identifica uma pessoa*.

Esse tipo de anomalia é de difícil reconhecimento, só podendo ser sanado pelo uso concorrente de um dicionário de relacionamento, ou seja, de uma fonte textual que descreve o relacionamento tanto quanto sua interpretação, como seu sentido de leitura. Dentre outras finalidades este, talvez, seja o principal papel do dicionário de relacionamentos.

- **Denominação do relacionamento com representação simultânea dos dois tempos verbais:**



Nesse caso, a posição relativa de um ou outro elemento formador do relacionamento (PESSOA e CARRO), sendo mais à esquerda ou acima, é indiferente no arbitramento das denominações. O que se deve ter como regra é que o termo de denominação mais próximo do elemento formador deve ser aquele que dê sentido de leitura *a partir deste elemento para o outro*. Assim, no exemplo anterior teríamos nos dois casos: "CARRO transporta PESSOA" e "PESSOA é transportada por CARRO".

Caracterização dos Relacionamentos

Como já foi visto, um relacionamento demonstra uma associação entre objetos, de igual ou diferentes tipos. Esses relacionamentos têm, todavia, características diversas que devem ser observadas e mapeadas. Caso não sejamos capazes de reconhecer todas essas características estaremos, apesar de identificando o relacionamento, criando um modelo sem as informações necessárias do ponto de vista conceitual.

Perceba que estamos, ainda, voltados aos aspectos conceituais e semânticos. Já nessa fase qualquer falta de precisão quanto ao reconhecimento das características dos objetos fará com que nosso modelo esteja errado. Mais a frente, veremos que essas imperfeições irão se propagar pelo modelo lógico e pelas estruturas de dados (tabelas) que venhamos a implementar. Isso fará com que não só conceitualmente tenhamos riscos de falsos entendimentos, mas que também na implementação de um sistema de informações que se baseie nesse modelo tenhamos resultados que não atendam às necessidades reais dos sistemas.

Assim, fica claro que, apesar de parecer extremamente simples, a fase de caracterização dos relacionamentos é um dos pontos vitais para a construção de modelos íntegros. Além da identificação dos relacionamentos já discutida na seção anterior, sua caracterização deverá ser feita baseada no atendimento de alguns requisitos:

1. Grau, ou cardinalidade, do relacionamento.
2. Número de elementos que participam do relacionamento.
3. Condição de participação dos elementos no relacionamento.
4. Condição de estabelecimento do relacionamento.

Cada um desses requisitos será tratado de modo isolado, em termos de conceituação e exemplificação, mas todos eles deverão ser vislumbrados de modo integrado durante o processo de caracterização do relacionamento.

Grau ou Cardinalidade do Relacionamento

Durante o processo de reconhecimento e entendimento de um relacionamento, estaremos sempre nos defrontando com regras de estabelecimento das associações entre os elementos. Essas regras não serão arbitradas mas sim extraídas do próprio ambiente observado. Em muitos casos, uma regra que seja verdadeira em um ambiente X observado poderá não ser verdadeira em outro ambiente Y.

Quando observado um relacionamento qualquer, deveremos nos fazer as seguintes perguntas para estabelecer o grau das associações:

Dado um relacionamento R entre elementos do tipo A e do tipo B:

1. Com quantos elementos do tipo B se relaciona *cada um dos* elementos do tipo A?
2. *Dado um* elemento do tipo B, com quantos elementos do tipo A ele se relaciona?

Imagine agora que os elementos A, B e R recebam os nomes:

Elemento A	Elemento R	Elemento B
PESSOA	DIRIGE	AUTOMÓVEL
PROJETO	É DESENVOLVIDO POR	ENGENHEIRO

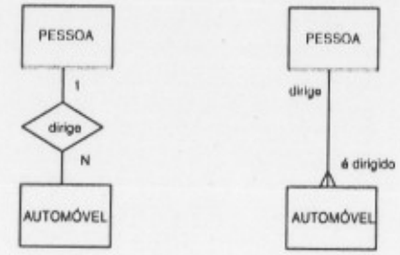
Fazendo cada uma das perguntas anteriormente mencionadas, teremos a definição do grau de cada um desses relacionamentos:

P.: Quantos AUTOMÓVEIS dirige *cada uma* das PESSOAS?
R.: *Uma* PESSOA *dirige vários* AUTOMÓVEIS, um de cada vez em nosso caso.

P.: Dado *um* AUTOMÓVEL, por quantas PESSOAS ele pode ser dirigido?
R.: *Um* AUTOMÓVEL *é dirigido por uma* única PESSOA em nosso caso.

As respostas às perguntas feitas em nossa investigação do ambiente observado deverão ser então mapeadas e representadas no modelo a ser criado. Essas "regras do negócio" deverão estar mapeadas para que, tanto para fins semânticos e conceituais, como para fins de criação futura de estruturas de dados, possamos ter a referência correta do ambiente observado.

Nesse caso, poderíamos gerar a seguinte representação:



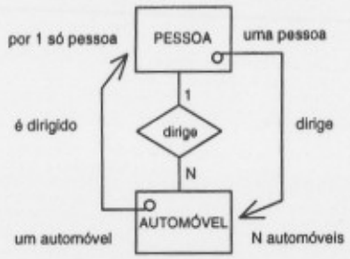
Notação Peter Chen Notação James Martin

Processo para a Definição e Notação de um Relacionamento

Perceba que junto ao elemento que representa o relacionamento (o losango na notação de Chen e a linha na notação de Martin), estaremos representando o grau de associação, ou cardinalidade. Para que se faça a correta alocação e leitura da cardinalidade da relação, devemos realizar as seguintes operações:

1. Fixar *um* elemento fictício pertencente ao conjunto de elementos do tipo PESSOA e perguntar a ele: Uma PESSOA pode dirigir quantos AUTOMÓVEIS? A resposta a essa pergunta será anotada no outro extremo do relacionamento.
2. Fixar *um* elemento fictício pertencente ao conjunto de elementos do tipo AUTOMÓVEL e, desconsiderando o resultado da operação anterior, perguntar: Um AUTOMÓVEL pode ser dirigido

por quantas PESSOAS? A resposta a essa segunda pergunta será anotada no outro extremo do relacionamento.



Note que mesmo tendo sido definido que "uma PESSOA dirige N AUTOMÓVEIS" *não devemos* analisar o lado contrário do relacionamento como "N AUTOMÓVEIS são dirigidos por uma PESSOA", *mas sim*, "um AUTOMÓVEL é dirigido por uma PESSOA". Esse tipo de confusão na interpretação e leitura de um relacionamento estabelecido é comum no início para aqueles que ainda não tenham fixado bem a seguinte regra:

> *Sempre o ponto de partida para a análise e leitura do relacionamento descrito deve ser* **um elemento individualizado.**

Possibilidades de Estabelecimento de Graus de Relacionamento

De acordo com as regras apresentadas, deveremos enquadrar os relacionamentos analisados em três grandes grupos, os quais, para fins de modelagem, cumprem a total finalidade de expressar a semântica de associação entre os elementos. As três alternativas de classificação são:

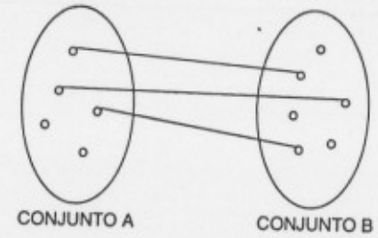
- 1:1 (leia-se Um para Um)
- 1:N (leia-se Um para Muitos)
- M:N (leia-se Muitos para Muitos)

Conceituação de Relacionamentos 1:1

Esse tipo de relacionamento denota um caso especial de um relacionamento 1:N onde, no caso, o valor de N é fixo e obrigatoriamente igual a 1. Esse tipo de grau de relacionamento é bastante restritivo na associação dos elementos e, por isso, acaba sendo pouco comum no dia-a-dia.

Como vimos pelo processo de obtenção e representação dos graus dos relacionamentos, um relacionamento de grau 1:1 denota que um elemento do tipo A só se relaciona com um elemento do tipo B e que, também, um elemento do tipo B só se relaciona com um elemento do tipo A.

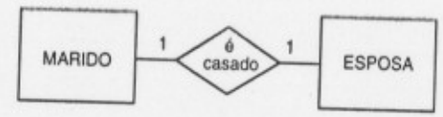
Genericamente, se considerarmos dois conjuntos de elementos entre os quais possa existir algum tipo de relacionamento, poderíamos representar esse tipo de associação como sendo:



Pode-se perceber claramente que a cada elemento do conjunto A somente um elemento do conjunto B é associado e vice-versa. Ainda neste instante não estaremos exigindo a associação de todos os elementos do conjunto A a pelo menos um elemento do conjunto B, ou que nenhum elemento do conjunto B permaneça sem associação a um elemento do conjunto A. Esse conceito será explorado mais a frente. Por hora, basta-nos estabelecer que, caso haja o relacionamento entre elementos, ele será de grau 1:1.

Exemplos:

pois uma PESSOA recebe uma é só uma CERTIDÃO DE ÓBITO, e, por sua vez, uma CERTIDÃO DE ÓBITO só pertence a uma PESSOA.



pois um MARIDO só é casado com uma ESPOSA, e, também, uma ESPOSA só estará casada com um MARIDO.

Os relacionamentos de grau 1:1 são bastante difíceis de serem caracterizados, pois qualquer mudança de interpretação ou visão pode fazer com que facilmente eles sejam questionados e até reconsiderados. Nos próprios exemplos dados, alguém poderia levantar pontos tais como:

Perceba que também no caso dos relacionamentos 1:N teremos que ter claramente observadas e mapeadas as regras de associação entre os elementos. Pequenas mas diferentes interpretações poderão fazer com que um relacionamento do tipo 1:N possa evoluir para um relacionamento M:N. No último exemplo dado, o que aconteceria se:

1. Uma turma viesse a ter uma professora assistente durante um certo número de aulas em que fosse executada uma atividade especial que demandasse atenção de mais de uma orientadora?

2. Uma turma tivesse uma professora no primeiro bimestre e outra no restante do período em função de troca de conteúdo programático, ou especialidade de ensino?

No caso de migração de um relacionamento 1:N para M:N, além das transformações semânticas que isso implica, teremos também mudanças radicais nas estruturas de armazenamento de dados caso, no futuro, pensemos em utilizar esse modelo para construir uma base de dados. Assim, é bastante importante que se estabeleça corretamente o grau de associação entre os elementos. Convém frisar que, em função de uma possível evolução do relacionamento, pode ser importante prever já esse relacionamento com o grau M:N, entretanto esta não deve ser uma regra. Muitos relacionamentos são e serão sempre do tipo 1:N. Mapeá-los como M:N, apesar de possível logicamente, seria semanticamente incorreto.

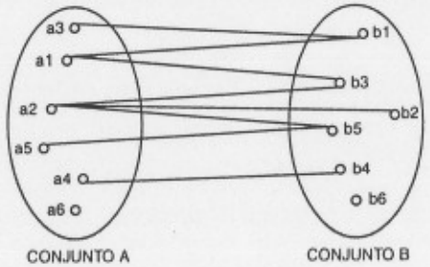
Conceituação de Relacionamentos M:N

Um relacionamento entre objetos que apresente o grau M:N representa uma associação de elementos onde, teoricamente, não há restrição quanto às possíveis ligações a serem estabelecidas entre os conjuntos A e B. Quando nos referimos à não-restrição entre as associações estamos, na verdade, nos referindo às restrições já apresentadas nos relacionamentos de grau 1:1 e 1:N. Num relacionamento M:N, por definição, um elemento qualquer de um conjunto A pode se associar a 0 (nenhum), 1 (um) ou N (vários) elementos do conjunto B. Por sua vez, um elemento qualquer do conjunto B pode se associar, também, a 0 (nenhum), 1 (um) ou N (vários) elementos do conjunto A.

Em termos de estrutura genérica de representação de elementos, podemos considerar um relacionamento de grau M:N como uma associação matricial, onde nas colunas temos os elementos do conjunto A e nas linhas os elementos do conjunto B. Cada uma das interseções, neste caso, denota um possível relacionamento entre A e B.

	a1	a2	a3	a4	a5	a6
b1	x		x			
b2		x				
b3	x	x				
b4				x		
b5		x			x	
b6						

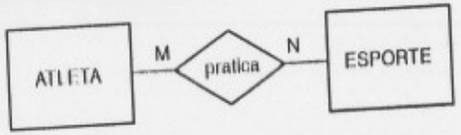
Caso observássemos os dois conjuntos A e B sob outra representação, teríamos que a cada elemento do conjunto A associaríamos N elementos do conjunto B que, por sua vez, poderiam estar associados a vários elementos no conjunto A. Assim, a matriz anteriormente utilizada para demonstrar as associações entre os elementos do conjunto A e B poderia ser vista como:



Perceba que não há obrigatoriedade de que todos os elementos do conjunto A tenham vários elementos do conjunto B associados a ele. Alguns podem não ter nenhum, outros ter somente um, e outros terem vários. O mesmo fato é valido com relação aos elementos do conjunto B. O que define o grau M:N em um relacionamento é o número máximo de associações, ou seja, podemos ter até N elementos associados, mas devemos lembrar que 0 e 1 também pertencem ao intervalo de números N válidos, que teoricamente varia de zero a infinito.

Outra consideração importante a ser lembrada é o porquê da denominação de um relacionamento desse gênero como M:N e não M:M. Utilizam-se diferentes letras para que explicitamente fique denotado que as duas grandezas são totalmente independentes. Isso significa que, por exemplo, se um elemento do conjunto A pode se relacionar com até 10 elementos do conjunto B, um elemento do conjunto B, não necessariamente, precisará também se relacionar com até 10 elementos do conjunto A. Um elemento do conjunto B poderia se relacionar com, no máximo, somente 2 elementos do conjunto A. Nesse caso, teríamos M=2 e N=10.

Exemplos:



pois um ATLETA pode dedicar-se à prática de mais de uma modalidade esportiva (por exemplo, 2) e também um dado esporte pode ser praticado por diversos atletas.

pois um certo livro pode ser escrito por um conjunto de autores (3, por exemplo), e, por sua vez, cada um dos autores pode escrever mais de um livro.

Como pode-se perceber, o grau de liberdade na associatividade entre os elementos é bastante grande. Isso faz com que, em alguns casos, critérios de garantia de integridade sejam adotados para inibir associações não desejadas. Esses aspectos serão discutidos mais detalhadamente quando estivermos tratando de aspectos lógicos da construção de modelos.

Mudanças na Notação de Relacionamentos M:N

Algumas notações acabam por induzir o modelador para que, se o relacionamento for de grau 1:1 ou grau 1:N, seja utilizada uma representação e se o grau do relacionamento for M:N seja utilizada outra. Isso deve-se ao fato de essas notações estarem bastante orientadas aos modelos lógicos e não aos modelos conceituais. No modelo lógico, como veremos mais adiante, será necessário criar uma estrutura de dados para implementar o relacionamento M:N, principalmente quando o relacionamento tiver atributos. Entretanto, não devemos deixar que restrições do modelo lógico impactem a criação do modelo conceitual.

Por isso, defendemos a idéia de que notações como a proposta por Martin são restritivas para a representação de modelos conceituais com maior grau de semântica. Em alguns casos, ela pode induzir a mudanças no modelo conceitual simplesmente por inabilidade na representação de determinadas situações.

Número de Elementos Que Participam do Relacionamento

Até agora, durante os exemplos apresentados, estivemos, para fins de facilitação de entendimento, utilizando a idéia de que um relacionamento se processa sempre entre dois objetos, sejam de tipos iguais ou diferentes.

A partir deste momento, já tendo sido demonstrada a possibilidade de aplicação desse tipo de abordagem no mapeamento de relacionamentos, iremos expandir o conceito de relacionamento. Na verdade, um relacionamento pode se estabelecer entre *dois ou mais elementos*, e não somente entre, no máximo, dois elementos.

Do ponto de vista conceitual, o estabelecimento de relacionamentos ternários, quaternários ou de maiores graus não é muito freqüente. Na maior parte das vezes, esse tipo de relacionamento pode estar mascarando um objeto não percebido, o qual concentra sobre si relacionamentos com os demais. Todavia, isso não deve ser estabelecido como uma regra, ou seja, nem sempre a ocorrência de um relacionamento não binário indica a existência de um objeto não percebido. Muitas vezes, o relacionamento será realmente entre mais de dois elementos e isso assim deverá ser mapeado.

Algumas notações, ou representações gráficas, fazem com que, obrigatoriamente, um relacionamento entre mais de dois objetos seja mapeado como uma entidade no modelo. A representação proposta por Peter Chen, neste instante, mostra-se mais flexível permitindo que se represente, sem grandes necessidades de ajustes, um relacionamento não binário. Já, por exemplo, a representação proposta por James Martin requer a criação de uma estrutura do tipo "entidade" para representar esse relacionamento.

Exemplos:

1. Relacionamentos Binários:

AVIÃO *transporta* CARGA
HOTEL *localiza-se em* CIDADE
REVISTA *contém* ANÚNCIO

2. Relacionamentos Ternários:

FUNCIONÁRIO *trabalha em* PROJETO *executando* FUNÇÃO
PEÇA *é fornecida por* FORNECEDOR *para um* PROJETO

Perceba que nos relacionamentos ternários somente as associações de elementos dois a dois não mapeiam de forma correta a informação desejada. Sempre que se queira a informação completa deverão ser agregados os três elementos. Muitos autores têm dito de modo simplista que um relacionamento ternário pode ser convertido em dois relacionamentos binários, mas isso deve ser considerado com todo o cuidado, pois dependendo do tipo de derivação não obteremos o resultado desejado. Na verdade, a estrutura necessária para a derivação de um relacionamento ternário é a AGREGAÇÃO. Essa nova estrutura, incorporada ao diagrama de entidade-relacionamento por extensões propostas ao modelo original, realmente resolve de modo correto o caso dos relacionamentos ternários e será vista mais adiante em nossa abordagem.

Para que se perceba melhor qual a realidade de um relacionamento ternário e sua impossibilidade de simples substituição por dois relacionamentos binários, vamos explorar um dos exemplos anteriormente citados.

Exemplo:

Suponha o caso em que um FUNCIONÁRIO de uma empresa prestadora de serviços seja alocado em diversos PROJETOS, com a condição especial de que em cada um dos projetos ele possa vir a exercer uma FUNÇÃO diferente. Assim, poderíamos ter:

Funcionário	Projeto	Função
Antônio Carlos	Recursos Humanos	Analista de Sistemas
Antônio Carlos	Adm. Financeira	Consultor
Antônio Carlos	Adm. Materiais	Analista de Sistemas
Carla Martins	Recursos Humanos	Programador
Carla Martins	Adm. Tráfego	Usuário
Marcos Silveira	Adm. Materiais	Consultor

Esses exemplos demonstram que um mesmo FUNCIONÁRIO pode exercer mais de uma FUNÇÃO desde que em diferentes PROJETOS. Também podemos ver que a um PROJETO são alocados diferentes FUNCIONÁRIOS, cada um com suas funções.

Pois bem, a pergunta que fica é: conseguiríamos mapear essa realidade fazendo associações desses elementos dois a dois? Vejamos:

Suponha que tentemos mapear os pares:

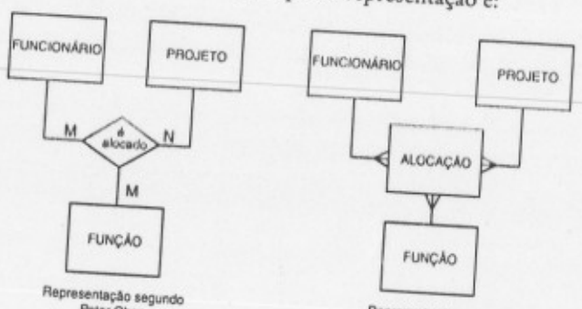
FUNCIONÁRIO — PROJETO
PROJETO — FUNÇÃO

Nesse caso, conseguiríamos definir, por exemplo, que Antônio Carlos está associado aos projetos Recursos Humanos, Adm. Financeira e Adm. Materiais. E também que no projeto Recursos Humanos temos estas funções sendo executadas: Analista de Sistemas e Programador.

Entretanto, juntando essas informações não somos capazes de definir quem é Analista de Sistemas em que projeto, ou com que função está alocada uma pessoa a um projeto.

Sabemos que Antônio Carlos está alocado em três projetos. Sabemos que nesses projetos existem diversas funções, mas qual é a função específica que Antônio Carlos executa no projeto Recursos Humanos? Analista de Sistemas ou Programador?

Outras combinações de elementos dois a dois poderão ser tentadas e sempre resultarão em informação parcial. Se quisermos expressar a FUNÇÃO de um FUNCIONÁRIO em um determinado PROJETO precisaremos em algum instante ter esses três elementos associados simultaneamente. Um modo de fazer esse tipo de representação é:



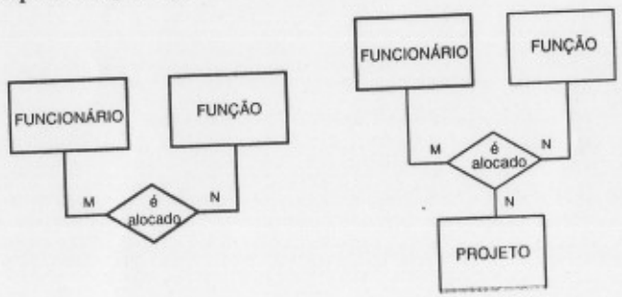
Representação segundo Peter Chen

Representação segundo James Martin

Pode-se perceber, no exemplo anterior que, conceitualmente, a notação de Chen é mais apropriada pois, se um dia, um relacionamento, que até então era percebido como binário, passar a ser ternário, bastará acrescentar esse novo elemento ao relacionamento sem que seja necessário mudar a própria estrutura de representação.

Assim, se o relacionamento antes percebido era de que um FUNCIONÁRIO podia exercer várias FUNÇÕES sem que fosse necessário definir em que PROJETO iria acontecer, poderíamos ter um modelo onde somente FUNCIONÁRIO e FUNÇÃO estivessem relacionados. Com novas necessidades sendo levantadas e novos mapeamentos sendo realizados, teríamos em um certo instante a necessidade de expandir nosso relacionamento binário transformando-o em ternário. Isso acabaria por ser traduzido da seguinte maneira nas duas notações básicas:

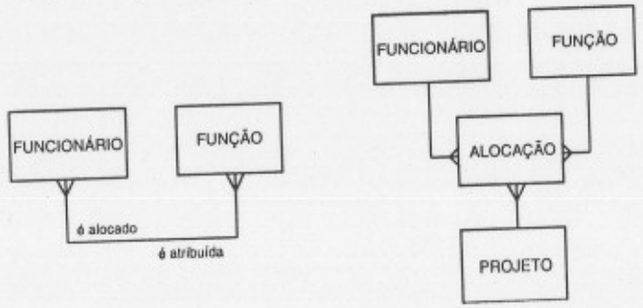
- **Representação segundo Peter Chen**



situação inicial

situação futura

- **Representação segundo James Martin**



situação inicial

situação futura

Outro tipo de conceito que deve ser manipulado quando se tratam os relacionamentos entre os objetos é a obrigatoriedade ou não da participação dos elementos nas associações passíveis de serem estabelecidas. Assim temos que, por opção, alguns elementos de um dado conjunto A podem não participar em nenhuma associação com elementos de um conjunto B. Do mesmo modo, qualquer um dos elementos do conjunto B pode não estabelecer associação com algum elemento do conjunto A.

Desse modo, durante o nosso trabalho de mapeamento de um relacionamento entre objetos, devemos, além de caracterizar os objetos, definir a semântica do relacionamento e definir o grau do relacionamento, também explicitar se serão permitidos, ou não, elementos em um ou outro conjunto, que não estabeleçam associações entre si. Veremos que em alguns casos o estabelecimento do relacionamento será obrigatório para todos os elementos e que, em outros, poderá ser opcional. O que definirá um ou outro caso será o ambiente observado e as regras de negócio estabelecidas entre os elementos.

Exemplo:

Nesse relacionamento, podemos reconhecer vários elementos úteis ao entendimento do relacionamento mapeado:

- Sabemos que existe uma escola.
- Sabemos que existem alunos.
- Reconhecemos que "uma escola atende a vários alunos".
- Reconhecemos que "um aluno é atendido por uma escola".

Mas, e quanto às perguntas adicionais que poderiam nos ser feitas:

- Toda escola tem que obrigatoriamente atender pelo menos a um aluno?
- Alguma escola pode não atender a aluno algum?
- Um aluno deve obrigatoriamente ser atendido por uma escola?
- Algum aluno pode não ser atendido por escola alguma?

Perceba que, para as perguntas apresentadas, o modelo mapeado não apresenta respostas. Pior que isso, as respostas podem depender do ambien-

te observado e mapeado. Para uma pessoa que imagine uma situação, as respostas podem ser uma, e para outra pessoa, com idéias e visões diferentes, as respostas podem ser outras.

Precisamos de algum modo enriquecer nosso modelo fazendo com que ele deixe de apresentar possibilidades de diferentes interpretações. Precisamos aumentar a semântica de nosso modelo fazendo com que às respostas às perguntas feitas não dependam mais de interpretação pessoal mas sim de observação do modelo. Para esse fim, será utilizado o conceito de relacionamentos condicionais e incondicionais.

Relacionamentos Incondicionais

Nos relacionamentos incondicionais não será denotado nenhum tipo de opcionalidade quanto à participação de elementos nas associações. Nesse tipo de representação precisaremos de informações adicionais para reconhecer se o relacionamento sempre segue o grau descrito (1:1, 1:N, M:N), ou se, por omissão, existem regras de exceção não qualificadas.

Os relacionamentos apresentados nos exemplos, até o momento, seguiam esse tipo de representação, não por ser esta a mais adequada, mas para fins de facilitação do entendimento. Esse recurso foi utilizado para possibilitar o progressivo aumento de detalhamento do assunto e assegurar uma melhor fixação dos conceitos básicos. Ressaltamos, fortemente, que esse tipo de representação tem baixo valor semântico e deve ser evitado para fins de modelagem conceitual.

Relacionamentos Condicionais

A representação dos relacionamentos condicionais garante ao leitor do modelo que a interpretação do modelo possa ser a mais fiel possível. Nesse tipo de representação estaremos, graficamente, demonstrando quando um ou outro elemento permite a ausência de associações.

Para esse fim deveremos introduzir mais um conceito importante na qualificação de um relacionamento. Até agora já havíamos visto o conceito de grau do relacionamento. A partir de agora, deveremos expandir esse conceito e pensar em dois tipos de grau: mínimo e máximo.

O que denominamos de grau máximo para fins de detalhamento, nada mais é do que o conceito até agora utilizado, ou seja, a determinação de relacionamentos como associações 1:1, 1:N e M:N. O que estaremos introduzindo como novo conceito será o grau mínimo, ou seja, qual é o menor valor possível de participação dos elementos do conjunto A e B no relacionamento.

Assim, um relacionamento, por exemplo, de grau 1:N, tal como o de "ESCOLA atende a ALUNOS", poderá ser qualificado como 1,1:1,N. Graficamente, teríamos:

Semanticamente, poderíamos interpretar esse relacionamento como:

Uma ESCOLA atende de 1 a N ALUNOS e
Um ALUNO é atendido por 1 e sempre 1 ESCOLA

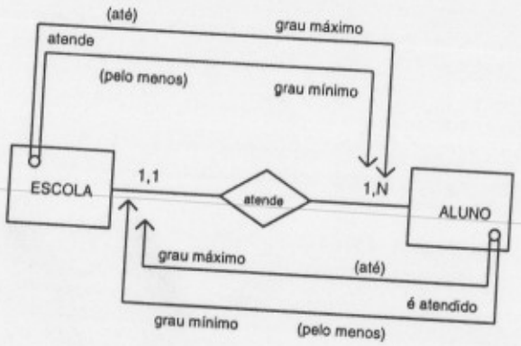
Ou através de outro modo de leitura:

Uma ESCOLA deve atender a pelo menos 1 ALUNO e até N ALUNOS
Um ALUNO deve ser atendido por uma e só uma ESCOLA

Ou, ainda:

Uma ESCOLA obrigatoriamente atende de 1 a N ALUNOS
Um ALUNO obrigatoriamente é atendido por 1 ESCOLA

Todas essas interpretações são feitas baseadas na observação das seguintes características:



Assim, nesse tipo de representação, estaremos sempre partindo de um dado elemento do conjunto A (ESCOLA) e definindo junto ao conjunto B (ALUNO) se o grau mínimo é 0 (nenhum aluno), 1 (um aluno), ou N (vários alunos). Para o exemplo dado, então, pode-se perceber que as respostas às perguntas antes feitas serão:

P.: Toda escola tem que obrigatoriamente atender a pelo menos um aluno?
R.: SIM, grau mínimo é 1 (1,1:1,N)

P.: Alguma escola pode não atender a aluno algum?
R.: NÃO, grau mínimo é 1 (1,1: 1,N)

P.: Um aluno deve obrigatoriamente ser atendido por uma escola?
R.: SIM, grau mínimo é 1 (1,1:1,N)

P.: Algum aluno pode não ser atendido por escola alguma?
R.: NÃO, grau mínimo é 1 (1,1:1,N)

Caso o ambiente por nós observado fosse outro, poderíamos ter diferentes casos e representações:

- Uma ESCOLA pode atender de Ø a N ALUNOS, mas todo aluno deve ser atendido por uma ESCOLA (1,1:0,N).
- Uma ESCOLA pode atender de Ø a N ALUNOS, mas nem todo ALUNO será atendido por uma ESCOLA (0,1:0,N).
- Uma ESCOLA deve atender de 1 a N ALUNOS, mas nem todo ALUNO será atendido por uma ESCOLA (0,1:1,N).

Pode-se perceber, portanto, que toda vez que nos interesse qualificar a opcionalidade de participação de um elemento em um relacionamento, o grau mínimo a ser atribuído à participação na associação será 0. Por outro lado, sempre que queiramos denotar a participação obrigatória de um elemento nas associações, devemos utilizar como grau mínimo o valor 1.

0 indica a possibilidade de não-participação ou opcionalidade.
1 indica a obrigatoriedade de participação.

Para fins semânticos, eventualmente podemos utilizar como grau mínimo, para denotar a obrigatoriedade, valores diferentes de 1. Nesse caso, estaremos sendo estritamente precisos quanto às faixas de valores aceitáveis em uma associação, o que por um lado pode ser bastante interessante para caracterizar a real situação mapeada, mas que, por outro lado, poderá tornar o modelo muito suscetível a mudanças. Esse tipo de notação é bastante infreqüente, apesar de semanticamente correta, portanto, pode ser utilizada mas, com certeza, causará bastante estranheza aos observadores de um modelo que siga esse padrão.

Exemplo:

denota que uma ESCOLA atende a no mínimo 20 ALUNOS e no máximo 100 ALUNOS, logo é uma associação obrigatória.

Outra possibilidade de notação de grau mínimo e máximo, já mais freqüentemente observada em modelos com características de reforço de semântica, é aquela onde o grau mínimo é expressado baseado nos conceitos de 0 e 1, como visto anteriormente, mas onde a caracterização do grau máximo é feita através da valoração do valor N ou M. Assim, pode-se saber se a relação é opcional (grau mínimo = 0) ou obrigatória (grau mínimo = 1), além de explicitar que, quando acontece a associação, ela se processa com um limite máximo conhecido.

Exemplo:

denota que uma ESCOLA atende a no mínimo 1 ALUNO, logo sendo obrigatória a associação, e no máximo N ALUNOS, sendo N conhecido como igual a 100.

Esses tipos de notação também são válidos para relacionamentos com grau diferente de 1:N, como os vistos nos exemplos anteriores. Aplicam-se perfeitamente a relacionamentos M:N onde, tanto de um lado como de outro, o grau mínimo e máximo podem situar-se dentro de uma faixa de valores predeterminados. Cabe ressaltar a inaplicabilidade desse tipo de notação a relacionamentos de grau 1:1 onde, invariavelmente, os valores máximo e mínimo em ambos os lados da associação só podem assumir valores 0 e 1.

Exemplo:

Quanto à Existência Simultânea de Relacionamentos

Já vimos através de alguns exemplos dados que um objeto pode vir a estabelecer relacionamentos simultâneos com diversos outros. Esta característica leva-nos a uma situação na qual será preciso, para fins semânticos, definir quando a existência de uma associação impacta, ou impede, a existência de outra.

Dentro desse enfoque podemos caracterizar, basicamente, três tipos de associações existentes entre objetos:

- Relacionamentos independentes.
- Relacionamentos contingentes.
- Relacionamentos mutuamente exclusivos.

Relacionamentos Independentes

A classe de relacionamentos que se enquadram como do tipo independente é caracterizada por agrupar todo e qualquer relacionamento que possa vir a ser estabelecido sem que haja necessidade de avaliação simultânea de outro relacionamento. Tratam de conceitos diferentes, podendo ser estabelecidos tanto entre entidades diferentes como, também, entre as mesmas entidades. Um exemplo em que temos entidades diferentes pode ser: "CANAL DE TV exibe PROGRAMA" e "PROGRAMA é patrocinado por FORNECEDOR". Perceba que a existência da associação entre PROGRAMA e CANAL DE TV é totalmente independente da associação entre PROGRAMA e FORNECEDOR. Podemos ter PROGRAMAS sem patrocínio, que continuarão a ser exibidos nos CANAIS DE TV, e também PROGRAMAS com patrocínio que, por falta de horário, não serão exibidos em nenhum CANAL DE TV

Outro exemplo, dessa mesma classe de relacionamentos, poderia ser estabelecido entre CLIENTE e FORNECEDOR. Imagine que "um CLIENTE é visitado por vários FORNECEDORES" mas que, paralelamente, "um CLIENTE faz cotação de preços com vários FORNECEDORES". Temos aí dois tipos distintos de relacionamentos, estabelecidos entre as mesmas entidades mas totalmente independentes. Podemos ter FORNECEDORES sendo consultados para cotação de preços, tanto por terem

visitado o CLIENTE como por indicação de terceiros. Também podemos ter FORNECEDORES que mesmo tendo visitado um CLIENTE não sejam incluídos no processo de cotação de preços

Pode-se perceber que o fato de dois, ou mais, relacionamentos serem estabelecidos entre duas entidades de diferentes tipos não é suficiente para nos levar à conclusão de que eles sejam independentes. Por outro lado, relacionamentos entre as mesmas entidades também podem ser independentes. Essa mesma regra é aplicável quando falamos de relacionamentos com dependência entre si. Poderemos ter dependências entre relacionamentos de entidades diferentes ou iguais.

Normalmente, um bom indício para a identificação de relacionamentos independentes está na conceituação dos relacionamentos. Associações com finalidades diferentes têm grande chance de serem independentes, apesar de isso não caracterizar efetivamente uma regra. O primeiro, mas não único sinal, de que possa haver independência é, portanto, a própria denominação dada ao relacionamento, ou em última instância seu nome.

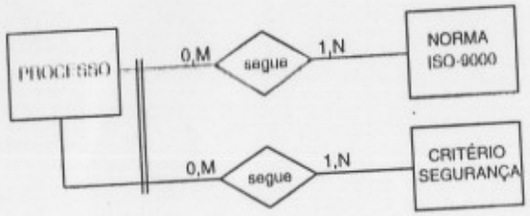
Lembre-se, entretanto, de que, nem sempre, nomes diferentes implicam independência dos relacionamentos, pois, por exemplo, "PRODUTO paga imposto à INSTITUIÇÃO" e "PRODUTO é isento por DECRETO-LEI" são dois relacionamentos distintos, com conceitos diferentes, nomes diferentes, e não são independentes. Nesse caso, existe uma mútua exclusão de associações. Sempre que PRODUTO relaciona-se com INSTITUIÇÃO não se relaciona com DECRETO-LEI e vice-versa. Veremos mais detalhes e exemplos de casos desse tipo quando tratarmos o caso de relacionamentos mutuamente exclusivos.

A abordagem mais eficiente para a identificação de independência entre relacionamentos é a interpretação da semântica do relacionamento e a análise de interdependência entre eles. Poderemos, inclusive, chegar a extremos onde, em um ambiente observado, um relacionamento possa ser considerado independente de outro, mas que, em se mudando o ambiente observado, este passe a criar vínculo de dependência com outro relacionamento.

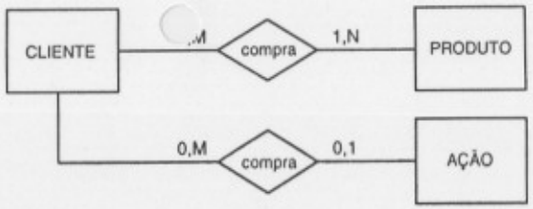
Relacionamentos Contingentes

Os relacionamentos que se enquadram na classe de contingentes são aqueles que, tendo dependência uns com os outros, impõem o estabelecimento simultâneo de associações entre os vários elementos envolvidos. Nesse caso, teremos estabelecido uma correlação entre dois ou mais relacionamentos através da análise da semântica de suas associações e, também, teremos definido que, como regra do negócio, mais de um relacionamento deve ocorrer em um mesmo instante.

Esse tipo de situação é bastante típico em associações que se processam com uma mesma finalidade, mas que envolvem elementos diferentes. Suponha que estejamos modelando um sistema de "Controle da Qualidade". Em um dado instante, identificamos que os PROCESSOS executados devem seguir NORMAS ISO-9000 e, também, CRITÉRIOS DE SEGURANÇA. Sabemos que as associações entre PROCESSO e NORMA ISO-9000 irão ocorrer concorrentemente às associações entre PROCESSO e CRITÉRIOS DE SEGURANÇA, mas, por serem elementos diferentes, não podemos agrupar todas essas associações como um só relacionamento. Assim, acabaremos por ter um modelo similar ao exibido abaixo:

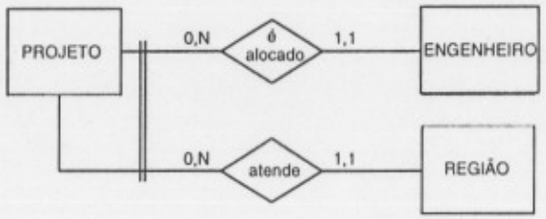


Há de se considerar, para fins de esclarecimento, que nem sempre onde obtivermos dois ou mais relacionamentos com o mesmo nome estaremos tratando de relacionamentos contingentes. Relacionamentos com diferentes nomes podem ser contingentes e relacionamentos com nomes iguais podem não ser. Como exemplo, podemos ter o caso onde "um CLIENTE compra um PRODUTO" e, paralelamente, "um CLIENTE compra AÇÕES da companhia". Perceba que o nome e o próprio significado do relacionamento "comprar" são similares mas que não podemos estabelecer como obrigatório o fato de que "todo CLIENTE que compre PRODUTOS também compre AÇÕES da companhia", ou vice-versa.



Relacionamentos não contingentes de nome igual

Como exemplo do caso contrário, onde relacionamentos de diferentes nomes e significados podem se caracterizar como contingentes, temos que "um PROJETO tem um ENGENHEIRO alocado ao ser iniciado". Paralelamente, e de modo contingente, este "PROJETO atende a uma REGIÃO da cidade". Perceba que nunca teremos um PROJETO com ENGENHEIRO que não atenda a uma REGIÃO da cidade. Podemos até definir, em instantes diferentes, as associações. Nesse caso, estaremos estabelecendo que, ao iniciar, o projeto já terá alocado a si um ENGENHEIRO e uma REGIÃO de atendimento.



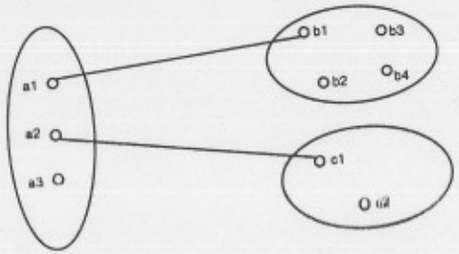
Relacionamentos contingentes de nomes diferentes

Aqui, mais uma vez, o ponto a ser observado para definição da contingência, ou não, entre os relacionamentos será a semântica das associações, ou, em outras palavras, o que elas representam efetivamente no modelo. Relacionamentos que devam ser estabelecidos sincronamente denotarão contingência, relacionamentos que possam ser estabelecidos opcionalmente denotarão independência.

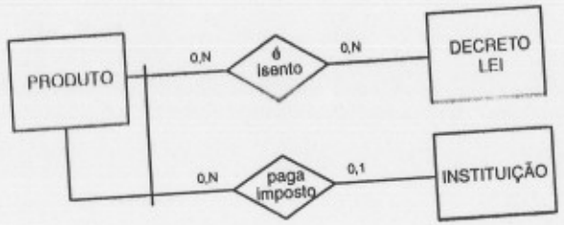
Relacionamentos Mutuamente Exclusivos

Dois, ou mais, relacionamentos estabelecidos entre os objetos de nosso modelo podem, em função de características próprias do ambiente observado, caracterizar-se por serem mutuamente exclusivos. Esse tipo de situação, como o próprio nome diz, implica que, se a associação de um elemento for estabelecida através de um dos relacionamentos, não poderá ser estabelecida pelos demais.

Algo importante a ser realçado é que a exclusividade se processa em nível de elemento individualizado dentro dos conjuntos de elementos de um ou outro tipo. Assim, se no conjunto A um elemento **a1** estabelece relação com um elemento **b1** do conjunto B, não poderá estabelecer relação com um elemento **c1** do conjunto C. Já outro elemento do conjunto A, por exemplo, a2, poderá estabelecer relação com **c1**, do conjunto C, e não com b2 do conjunto B. Deve ficar claro que não haveria sentido imaginarmos a exclusividade se processando para todo o conjunto A do mesmo modo, pois, neste caso, um dos relacionamentos nunca poderia ser estabelecido ficando assim sem significado prático.



Voltemos a um exemplo já citado para demonstrar esse tipo de relacionamento. Vimos, quando falávamos de relacionamentos independentes, que os relacionamentos "PRODUTO paga imposto à INSTITUIÇÃO" e "PRODUTO é isento por DECRETO-LEI" eram dois relacionamentos distintos, com conceitos diferentes, nomes diferentes, e não se enquadravam como independentes. Nesse caso, como vimos, existia uma mútua exclusão de associações. Sempre que PRODUTO se relacionasse com INSTITUIÇÃO não se relacionaria com DECRETO-LEI e vice-versa. Em termos de representação gráfica, teríamos o seguinte modelo:

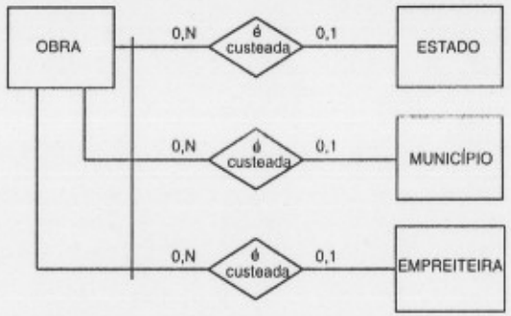


Relacionamentos mutuamente exclusivos de nomes diferentes

Perceba que, no exemplo dado, utilizamos conceitos e nomes completamente diferentes para cada um dos relacionamentos mutuamente exclusivos. Isso, no entanto, não é uma condição necessária para a ocorrência de relacionamentos mutuamente exclusivos. Poderíamos ter relaciona-

mentos com nomes iguais, entre entidades iguais, e com qualquer outro tipo de característica e, mesmo assim, dependendo do significado que expressassem, poderíamos enquadrá-los como mutuamente exclusivos.

Assim, no caso dos relacionamentos "OBRA é custeada por MUNICÍPIO", "OBRA é custeada por ESTADO" e "OBRA é custeada por EMPREITEIRA", poderíamos estabelecer que, em função das regras do ambiente observado, sempre que uma OBRA fosse realizada com fundos do MUNICÍPIO não poderia ser também custeada pelo ESTADO ou por uma EMPREITEIRA e vice-versa. Poderíamos até imaginar uma situação diferente onde tanto ESTADO como MUNICÍPIO custeassem uma mesma OBRA, mas não é esse o caso em nosso exemplo. Nessa situação, teríamos três relacionamentos com nomes e conceitos iguais mas mutuamente exclusivos entre si.



Relacionamentos mutuamente exclusivos de nomes iguais

Quanto à Presença de Atributos

Muitas vezes, durante o processo de modelagem, iremos nos defrontar com situações nas quais deveremos manipular informações (mais precisamente, dados) que pertençam não aos objetos do ambiente observado, mas à associação desses objetos. Isso fará com que seja necessário expandir nosso entendimento sobre o mapeamento de atributos e de relacionamentos.

De modo simplificado, como vimos até o momento, ao observarmos um ambiente, estaremos percebendo os objetos, seus atributos e seus relacionamentos. Entretanto, uma consideração adicional deve ser feita para que consigamos mapear completamente um ambiente. Muitos relacionamentos, ao serem estabelecidos através da associação de elementos individualizados, passam a conter dados, ou atributos, que devem então ser mapeados. Como isso não é uma regra que possa ser estabelecida como sempre verdadeira ou sempre falsa, passaremos a ter de tratar, para fins de classificação dos relacionamentos, aqueles *com atributos* e aqueles *sem atributos*.

Algumas metodologias e técnicas de representação gráfica desconsideram esse tipo de classificação utilizando o recurso de transformar todo relacionamento com atributo em uma entidade. Esse recurso tem uma influência muito forte do método de modelagem lógica e, muitas vezes, é imposto pela própria falta de elementos gráficos previstos para esse fim na metodologia de diagramação. Do ponto de vista conceitual, e caso nosso padrão de representação gráfica permita, devemos representar os atributos de um relacionamento no próprio relacionamento sem a necessidade de criar uma pseudo-entidade para acomodá-los.

Relacionamentos com Atributos

Um relacionamento com atributos tem todas as características comuns a qualquer outro relacionamento. Tem um nome, um significado, um grau mínimo e máximo, pode ser independente, contingente, mutuamente exclusivo, ou ter qualquer outro tipo de classificação já vista. O ponto importante a ser considerado é que, adicionalmente, a todos esses elementos, ele possui também dados a serem manuseados. Esses dados denotam a existência de informações que só podem ser estabelecidas ou consideradas quando da existência de uma associação entre elementos.

Muitas vezes, por simplificações ou falhas, os atributos de um relacionamento acabam por ser alocados em uma das entidades que o compõem. Isso pode ser válido em alguns casos, normalmente em relacionamentos de grau 1:1 e 1:N, mas nem sempre aplicável a relacionamentos M:N. Mais comumente as pessoas têm facilidade em determinar atributos para relacionamentos de grau M:N, o que nem sempre acontece com outros tipos de relacionamento. Entretanto, do ponto de vista conceitual, a correta alocação dos atributos deve procurar ser respeitada fazendo com que, se os atributos pertencem ao relacionamento, ali eles sejam alocados.

A identificação de local de apropriação dos atributos, se na entidade ou no relacionamento, deve ser feita em função de seus significados e instante de surgimento. Assim, se um atributo só pode ser determinado a partir do estabelecimento da associação entre dois elementos, ele deve pertencer ao relacionamento e, se o atributo existe mesmo antes de qualquer associação, ele pertence à entidade.

Veremos, através dos exemplos a seguir, que apesar de parecer simples esse tipo de interpretação deve ser feita com rigor para garantir a correta alocação e evitar vícios de projeto que comumente são praticados.

Imaginemos um primeiro caso onde "um ENGENHEIRO é alocado a vários PROJETOS" e "um PROJETO tem como responsável um ENGENHEIRO". Estamos frente a um relacionamento de grau 1:N, o que para muitas pessoas já indica que não devam existir atributos no relacionamento.

Entretanto, após o levantamento dos dados existentes no ambiente, poderíamos ter chegado a uma lista de atributos que deveriam ser alocados aos objetos e, eventualmente, ao relacionamento. Vamos investigar alguns atributos e identificar suas propriedades:

Atributo	Local de Alocação	Proprietário
nome do engenheiro	ENTIDADE	ENGENHEIRO
número da carteira de trabalho	ENTIDADE	ENGENHEIRO
especialidade	ENTIDADE	ENGENHEIRO
data de alocação ao projeto	RELACIONAMENTO	é alocado a
horário de trabalho no projeto	RELACIONAMENTO	é alocado a
código do projeto	ENTIDADE	PROJETO
duração prevista do projeto	ENTIDADE	PROJETO
custo do projeto	ENTIDADE	PROJETO

Perceba que dois atributos foram alocados no relacionamento. A justificativa para isso está no fato de que, por definição, "data de alocação ao projeto" e "horário de trabalho no projeto" não são atributos que possam existir previamente junto ao ENGENHEIRO ou ao PROJETO. Só saberemos a data de alocação de um ENGENHEIRO em um PROJETO quando essa associação se concretizar. Do mesmo modo, só saberemos qual será o horário de trabalho (diurno ou noturno) do ENGENHEIRO depois que tenhamos associado ele a um PROJETO.

Se em qualquer instante a associação não for mais válida e deixar de existir, os atributos também deverão deixar de ter significado. Considere que não estamos mapeando a questão do controle histórico das alocações, ou coisa do gênero, o que implicaria outro tipo de relacionamento. Normalmente, seríamos induzidos a considerar esses dois atributos como pertencentes ao PROJETO, já que um ENGENHEIRO pode ser alocado a mais de um PROJETO. Isso, apesar de válido no projeto lógico, não traduz conceitualmente a realidade observada.

Estaríamos mapeando desse modo uma situação onde, logo após um PROJETO ser criado, e mesmo sem que tivéssemos definido um ENGENHEIRO para ser alocado a ele, já teríamos os atributos "data de alocação ao projeto" e "horário de trabalho no projeto" associados ao PROJETO, o que sabemos é falso. Também estaríamos permitindo que mesmo um PROJETO que deixou de ter um ENGENHEIRO a ele alocado por

desistência do serviço tivesse os atributos "data de alocação [illegible] projeto" e "horário de trabalho no projeto".

O que, em princípio, pode causar certa estranheza para aqueles acostumados ao projeto lógico é o fato de um atributo não estar nem na entidade A nem na entidade B e sim em seu relacionamento. Devemos lembrar, mais uma vez, que o modelo conceitual serve a inúmeras outras finalidades, entre elas o projeto lógico de bancos de dados. Nesse último caso, teremos, ao seu instante, regras para a derivação das estruturas de dados necessárias para a implementação desse tipo de situação. Lá então veremos que, realmente, para fins de simplificação, poderemos migrar os atributos do relacionamento 1:N para dentro da entidade pertencente ao grau N. Mas é bom que fique claro que esta será apenas uma das alternativas dentre outras.

Quanto a atributos pertencentes a relacionamentos M:N, poderemos exemplificar utilizando um caso bastante conhecido e de fácil visualização. Vamos imaginar uma situação em que se queira mapear um controle acadêmico básico onde "ALUNOS freqüentam CURSOS" e "CURSOS são freqüentados por ALUNOS".

Em nosso caso, um ALUNO deve freqüentar pelo menos um CURSO, podendo freqüentar todavia vários CURSOS. Por seu lado, um CURSO é freqüentado por nenhum ou até vários alunos. Isso caracteriza, apesar dos graus mínimos, um relacionamento M:N.

Vejamos agora alguns possíveis atributos a serem manuseados dentro desse ambiente e suas propriedades.

Atributo	Local de Alocação	Proprietário
nome do aluno	ENTIDADE	ALUNO
endereço	ENTIDADE	ALUNO
data de nascimento	ENTIDADE	ALUNO
nome do curso	ENTIDADE	CURSO
carga horária	ENTIDADE	CURSO
nota para aprovação	ENTIDADE	CURSO
quantidade de faltas	RELACIONAMENTO	participa
média final	RELACIONAMENTO	participa
data de inscrição	RELACIONAMENTO	participa

Os atributos "nome do aluno", "endereço" e "data de nascimento" são bastante típicos e facilmente identificáveis como pertencentes ao ALUNO. Não interessa em que CURSO o ALUNO esteja inscrito, esses atributos serão sempre fixos e pertencentes ao ALUNO, pois ele é que tem um nome, mora em um endereço e nasceu em uma certa data.

Os atributos "nome do curso", "carga horária" e "nota para aprovação" também não causam muita dúvida quanto a sua propriedade. Mesmo que não saibamos quem ou quantos ALUNOS estão inscritos em um CURSO, os atributos desse curso existirão e serão fixos, pois o CURSO é quem possui um nome, uma nota final mínima para aprovação e uma carga horária.

Resta-nos agora investigar os atributos "quantidade de faltas", "média final" e "data de inscrição. Que situação poderíamos ter? Vamos imaginar alguns exemplos e verificar como se processa a alocação dos atributos no relacionamento. Poderíamos ter para o CURSO de "Análise de Sistemas" as inscrições de "João", "Pedro" e "Maria". "João" se inscreveu no dia 25 de março, "Pedro" no dia 30 de março e "Maria" no dia 29 de março. Cada um deles participou do treinamento durante todo o mês de abril e, ao final, no fechamento do boletim escolar, obtiveram as seguintes notas e faltas: "João" com 5 faltas obteve nota 8,5; "Pedro" com 2 faltas obteve nota 7,0 e "Maria" com 2 faltas obteve nota 8,0.

Perceba que a informação de que "Pedro" teve 2 faltas e média final 7,0 não são apropriáveis à entidade CURSO pois, se assim fosse, as faltas de "João" e Maria também deveriam estar lá, o que faria com que somente a última informação inserida fosse preservada. Também essas duas informações não poderiam ser apropriadas junto ao ALUNO, pois se "Pedro" fizesse simultaneamente dois CURSOS sempre perderia as informações de um deles.

Mas onde então colocar esse tipo de informação? A alternativa de colocar esses atributos no relacionamento entre as duas entidades criará a estrutura adequada para armazenamento dos dados necessários.

Sempre que estivermos tratando de atributos que sejam alocados em um relacionamento do tipo M:N, devemos tentar imaginar uma estrutura matricial onde nas linhas teremos os atributos de uma das entidades, nas colunas os atributos da outra entidade e nas células de interseção os atributos do relacionamento. Desse modo, poderíamos imaginar algo como:

Alunos	Cursos		
	Análise de Sistemas 32 horas/aula média 7,5	Programação Basic 80 horas/aula média 8,0	Lógica 32 horas/aula média 7,0
nome = João end. = Rua Augusta 100 nasc. = 11/07/64	inscrição = 25/03/95 faltas = 5 nota = 8,5		
nome = Pedro end. = Rua Floriano, 92 nasc. = 05/05/65	inscrição = 30/03/95 faltas = 2 nota = 7,0	inscrição = 22/09/95 faltas = 0 nota = 10	
nome = Maria end. = Av. Alves Lima, 223 nasc. = 19/12/64	inscrição = 29/03/95 faltas = 2 nota = 8,0		inscrição = 11/05/95 faltas = 8 nota = 6,7

Pela observação do quadro podemos constatar que realmente o grau de nosso relacionamento é M:N, pois um ALUNO (Maria) participa de mais de um CURSO (Análise de Sistemas e Lógica), tendo em cada um deles diferentes datas de inscrição, faltas e notas. Por outro lado, um CURSO (Análise de Sistemas) tem mais de um ALUNO participando (João, Pedro e Maria), cada um deles também com seus diferentes atributos.

Fica claro que os dados de nome do aluno, endereço e data de nascimento não precisam ser repetidos em cada uma das células de interseção (ou no relacionamento). Também assim não precisamos repetir os dados nome do curso, carga horária e média para aprovação em cada uma das células, bastando tê-los apenas uma vez em cada uma das colunas. Os únicos valores que não podem ser fixados, tendo que ser definidos célula a célula, são as datas de inscrição, notas e médias do aluno no curso, isso porque cada ALUNO em cada um dos CURSOS de que participe poderá ter valores diferentes para esses atributos.

Relacionamentos Sem Atributos

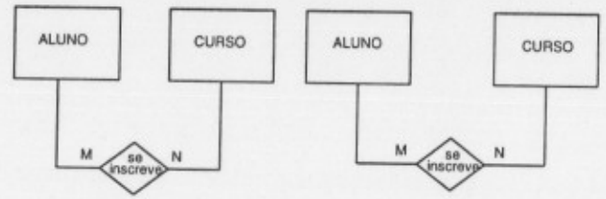
Em contrapartida a um relacionamento com atributos, um relacionamento sem atributos é aquele que se estabelece entre elementos de dois, ou mais, conjuntos e para o qual não são identificadas informações a serem mantidas ou mapeadas.

Os relacionamentos sem atributos podem pertencer a qualquer uma das classificações já vistas, como, por exemplo: grau 1:1, 1:N e M:N, relacionamentos independentes, contingentes ou mutuamente exclusivos e assim por diante. Esses tipos de relacionamentos serão mais freqüentes dentro da categoria de relacionamentos de grau diferente de M:N, o que não impede que também esteja presente nesse tipo de grau.

Futuramente, pela análise mais detalhada, ou pela própria evolução do modelo, um relacionamento que hoje seja sem atributos poderá evoluir para um relacionamento com atributos e vice-versa. Caso estejamos utilizando algum tipo de representação gráfica que faça distinção entre esses dois tipos de relacionamentos, deveremos fazer os devidos ajustes no modelo para que a representação seja fiel aos conceitos identificados. No caso a seguir, mostramos uma notação que implica a mudança de representação de um mesmo relacionamento entre as entidades ALUNO e CURSO simplesmente porque em um dado instante não haviam sido reconhecidos atributos necessários ao relacionamento e, futuramente, foram identificados tais atributos.

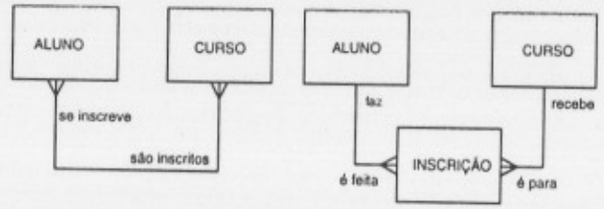
Mudanças na Representação em Função de Relacionamentos com Atributos

Outro motivo que tem levado os modeladores a já no nível conceitual terem de criar estruturas do tipo "entidade" para espelhar relacionamentos é o fato de o relacionamento conter atributos a serem mapeados. Na notação de Chen esse fato é irrelevante, pois tanto para um relacionamento com atributos como em um relacionamento sem atributos a notação será mantida. Já na notação proposta por Martin um relacionamento com atributos deve ser transformado em entidade, pois não há, em princípio, meio de alocar atributos sem que se tenha uma entidade para esse fim.



A) sem atributos Chen

B) com atributos Chen



A) sem atributos Martin

B) com atributos Martin

Esse caso, além de servir como exemplo para a necessidade de mudanças no modelo criado em função do surgimento de atributos em um relacionamento, reforça de modo claro que:

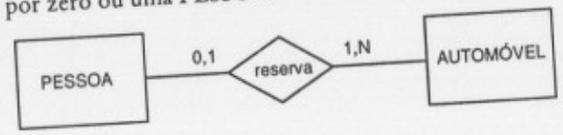
> *Entidades caracterizadas por interações (casamento, venda, inscrição...) podem ser substitutas para relacionamentos com atributos (casa com, vende para, se inscreve...) mantendo a mesma semântica através de outro componente.*

Já havíamos visto essa característica quando falamos de objetos-interação. Naquele instante já afirmávamos que sempre que um objeto-interação for identificado, ele poderá ser substituído por um relacionamento. O modelo a ser criado dependerá da representação gráfica que se esteja utilizando e da possibilidade ou não de representar uma estrutura de relacionamento com e sem atributos.

Dicionarização de Relacionamentos

Durante o processo de modelagem conceitual, freqüentemente grande ênfase é dada à dicionarização das entidades e seus atributos. Algumas ferramentas CASE sequer permitem que um relacionamento seja dicionarizado. Entretanto, para que realmente seja entendido o papel de um relacionamento, muitas vezes será necessário agregar algum tipo de informação adicional sobre o relacionamento estabelecido entre as entidades. O conhecimento da cardinalidade, utilizando-se graus mínimos e máximos, e a utilização de nomes significativos podem ajudar mas, freqüentemente, não são suficientes para o completo entendimento dos modelos.

Isso pode parecer, em princípio, pouco comum mas vejamos um exemplo. No modelo a seguir, temos um relacionamento denominado "reserva" estabelecido entre PESSOA e AUTOMÓVEL. Vemos que uma PESSOA pode reservar de 1, N AUTOMÓVEIS e que um AUTOMÓVEL pode ser reservado por zero ou uma PESSOA.



Mas o que isso significa?

- Que uma PESSOA *reserva* um AUTOMÓVEL para uma viagem em empresa de locação de automóveis?

- Que uma PESSOA *reserva* um AUTOMÓVEL para transportá-la de um local ao outro em uma empresa de táxis executivos?
- Que uma PESSOA *reserva* um AUTOMÓVEL para futura compra em uma concessionária de veículos?
- Ou quem sabe ainda outras alternativas possíveis ...

Perceba que todas essas alternativas poderiam ser válidas para o relacionamento "reserva". Como saber qual a interpretação correta? E, além disso, como saber por que um AUTOMÓVEL só pode ser reservado por uma PESSOA? E por que uma PESSOA tem que obrigatoriamente reservar pelo menos um AUTOMÓVEL?

As respostas a essas perguntas serão, normalmente, parte do conhecimento detido pelo responsável pela construção do modelo. Serão fatos e informações que ele terá obtido durante a observação do ambiente a ser modelado e que terão ficado sob seu poder. Porém, já vimos anteriormente que uma das finalidades do modelo conceitual (e dos elementos que o acompanham, tal como o dicionário de dados) é servir como meio para a representação do ambiente observado. Se não formos capazes de expressar totalmente o conjunto de informações obtidas, estaremos tendo algum tipo de deficiência.

Com isso, procuramos demonstrar que a dicionarização dos relacionamentos deverá ser um elemento adicional para a transmissão das informações obtidas durante a fase de modelagem e, adicionalmente, um meio para diminuir o grau de ambigüidade que possa vir a ser gerado pela simples "leitura" de um modelo.

> *E como seria então a dicionarização de um relacionamento? Que elementos deveriam ser observados e comentados?*

Dentre as principais informações a serem fornecidas para caracterizar um relacionamento de forma inequívoca, devemos ter:

- Sua função.
- O que ele representa.
- Quais são as regras de seu estabelecimento.
- Quais são as exceções a seu estabelecimento.
- Quando ele ocorre.
- Quando ele pode deixar de existir.

Assim, no exemplo anterior, quando citamos o caso de um relacionamento entre PESSOA e AUTOMÓVEL denominado "reserva", poderíamos ter gerado a seguinte dicionarização:

Representa as encomendas feitas através da rede de concessionárias onde um cliente, após ter seu crédito aprovado, paga 50% do valor do automóvel e entra em uma fila de espera para receber o bem desejado. Uma pessoa pode fazer até 3 reservas no máximo. Quando uma reserva for cancelada ela deverá continuar constando dos arquivos para fins de controle. As reservas canceladas continuam a ser contadas dentre as 3 possíveis reservas de um cliente.

Agora, observando-se o modelo e consultando-se o dicionário de dados, qualquer indivíduo que não tenha participado do processo de modelagem será capaz de entender a real finalidade do relacionamento "reserva". A ambigüidade terá sido reduzida praticamente a zero. Isso prova que, apesar de parecer pouco importante, ou muitas vezes parecer trabalhosa, a tarefa de dicionarização do relacionamento poderá ser imprescindível para o correto entendimento do modelo gerado.

Elementos de Caracterização Semântica Adicionais

Basicamente, qualquer modelo poderia ser construído com um mínimo suficiente de semântica baseado nos conceitos de entidades, atributos e relacionamentos. Essa era a proposta inicial de Peter Chen quando formalizou o Modelo Entidade-Relacionamento (MER). Acontece que, com o passar do tempo, novas características semânticas passaram a demonstrar ser necessárias para o completo entendimento dos modelos.

A partir dessas novas necessidades, alguns elementos adicionais passaram a ser agregados aos modelos de entidade-relacionamento. Hoje, vemos alguns deles fortemente inseridos no ambiente de modelagem orientada a objetos, passando, inclusive, a fazer parte obrigatória desses modelos. Já do ponto de vista de modelagem de dados, alguns desses elementos continuam a não ser freqüentemente utilizados, o que, pelo aspecto lógico, é possível, mas pelo aspecto conceitual acaba por deixar os modelos bastante deficientes.

Estrutura de Generalização e Especialização

Semanticamente, ao observarmos um universo qualquer, podemos reconhecer agrupamentos e distinções entre conjuntos de objetos identificados. Esse é o principal objetivo do processo de abstração e identificação dos objetos. Entretanto, devemos estar atentos para o fato de que nem sempre a separação de instâncias desses objetos, ou elementos, se dará por conjuntos estritamente distintos.

Caso tenhamos reconhecido conjuntos distintos, teremos identificado entidades ou objetos distintos. Mas, e no caso da identificação de subconjuntos distintos dentro de conjuntos únicos, como tratá-los? Vejamos alguns exemplos para caracterizar essas situações:

Imagine que ao observar um ambiente hospitalar, tenhamos reconhecido os seguintes elementos:

- MÉDICOS
- PACIENTES
- QUARTOS
- SALAS DE CIRURGIA

Fica evidente que cada conjunto desses possuirá diversos elementos associados a ele e isso nos denotará as entidades ou objetos de nosso modelo. Até aí não tivemos qualquer necessidade adicional de caracterização. Mas imagine que dentre os MÉDICOS, por uma observação mais detalhista, tenhamos reconhecido duas categorias distintas do tipo:

- MÉDICOS RESIDENTES
- MÉDICOS EFETIVOS

Cada uma dessas categorias tem, apesar de várias características comuns, características próprias distintas mutuamente aplicáveis caso a caso. Imagine, por exemplo, alguns atributos, tais como:

- Especialidade
- Nome do Médico
- Local de Atuação
- Tempo de Permanência
- Data de Efetivação
- Data de Início da Residência
- Nome do Orientador
- Data de Avaliação Prevista
- Tempo de Experiência na Função

Como cada um seria apropriado a um e outro tipo de médico? Todos poderiam ser alocados a todos os tipos de médicos? Como resolver semanticamente esse tipo de situação?

Do ponto de vista de modelagem, podemos contar com a estrutura de Generalização-Especialização, ou Gen-Espec, como utilizam alguns autores. Essa estrutura, como o próprio nome diz, procura representar graficamente o fato de que:

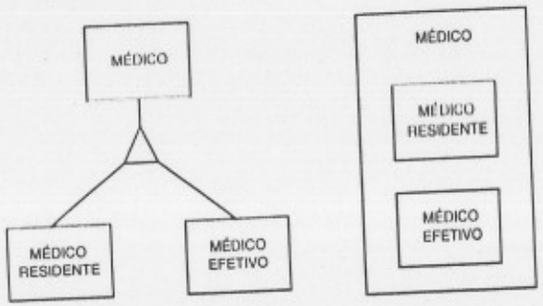
Dado um conjunto de elementos pertencentes a subconjuntos de um conjunto maior, cada um deles deve ser capaz de ser visto como um elemento tanto pertencente aos subconjuntos distintos como ao conjunto completo.

Assim, o médico residente de nome Alexandre pertence ao conjunto dos MÉDICOS RESIDENTES e também ao conjunto dos MÉDICOS em geral. Do mesmo modo, o médico efetivo Guilherme pertence ao subconjunto dos MÉDICOS EFETIVOS e também ao conjunto dos MÉDICOS em geral.

Devemos lembrar, porém, que somente para o médico Alexandre, que é um residente, teremos os atributos Data de Início da Residência, Nome do Orientador e Data de Avaliação Prevista como aplicáveis. E também, somente para o médico Guilherme, que é um efetivo, teremos os atributos Local de Atuação, Tempo de Permanência e Data de Efetivação. Mas, e o que dizer dos demais atributos: Especialidade, Nome do Médico e Tempo de Experiência na Função? Não se aplicariam, por acaso, esses atributos a quaisquer um dos tipos de médico? Sim, e assim devem ser modelados

Atributo	Médico Residente	Médico Efetivo
Especialidade	x	x
Nome do Médico	x	x
Local de Atuação		x
Tempo de Permanência		x
Data de Efetivação		x
Data de Início da Residência	x	
Nome do Orientador	x	
Data de Avaliação Prevista	x	
Tempo de Experiência na Função	x	x

Logo, para que essa situação possa ser representada em nossos modelos iremos introduzir um novo elemento semântico, e gráfico, que denota grupos e subgrupos:

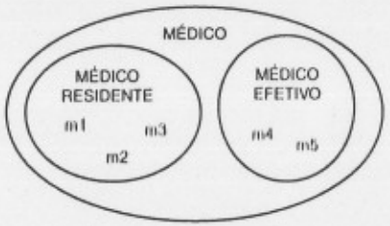


notação 1

notação 2

Através da notação 1, podemos observar que a entidade MÉDICO subdivide-se em duas outras. Devemos visualizar isso não como um conjunto de três entidades distintas mas como o triângulo sendo um prisma no qual, ao incidir a entidade MÉDICO, podemos ver refletir do outro lado seus dois espectros distintos: MÉDICOS RESIDENTES e MÉDICOS EFETIVOS. De um dos lados do triângulo temos o agrupamento total dos elementos e, do outro, os subconjuntos possíveis obtidos a partir da divisão do todo.

Já pela notação 2, talvez mais facilmente reconhecível, podemos observar a existência de um grande conjunto de elementos (MÉDICOS) que envolve todo o universo tanto do subconjunto dos MÉDICOS RESIDENTES como dos MÉDICOS EFETIVOS. Assim, podemos perceber que, sendo um MÉDICO RESIDENTE um elemento, também está inserido no conjunto dos MÉDICOS, e que cada MÉDICO ou estará no subconjunto dos MÉDICOS RESIDENTES ou, dos MÉDICOS EFETIVOS.



Apesar de graficamente distintas, as duas notações têm a mesma finalidade, a mesma riqueza de informação e a mesma possibilidade de uso semântico. A preferência pelo uso de uma ou outra representação pode estar vinculada à ferramenta de desenho utilizada ou a preferências pessoais. Veremos mais a frente, quando tratarmos especificamente o assunto de padrões de representação gráfica, que ainda outras possibilidades de representação podem ser observadas em outras metodologias. O importante será distinguir em cada uma delas a existência da característica de representação do conjunto global (a generalização) e dos subconjuntos (as especializações).

Perceba que as especializações, apesar do nome, não estão ligadas somente a funções especializadas executadas por pessoas. São simplesmente papéis especializados que um elemento toma dentro de um conjunto maior. Desse modo poderíamos ter vários outros exemplos para essa estrutura, tais como:

Generalização	Especializações
Veículo Automotor	Carro, Caminhão, Trator
Animal	Animal Doméstico, Animal Selvagem
Item de Estoque	Item perecível, item não perecível
Hardware	Disco Rígido, Disco Flexível, Monitor

Grande parte dos exemplos dados para demonstrar a estrutura de Gen-Espec tinha uma característica em comum: criavam subconjuntos mutuamente exclusivos entre si. Assim, se um MÉDICO pertencesse ao subconjunto dos MÉDICOS RESIDENTES não pertenceria ao dos MÉDICOS EFETIVOS. Se um item de HARDWARE fosse do subconjunto dos DISCOS RÍGIDOS não seria do subconjunto dos MONITORES.

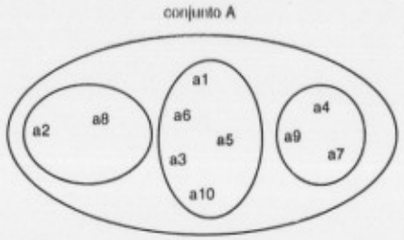
Perceba que apesar de um elemento estar simultaneamente presente no conjunto global e em um dos subconjuntos, ele nunca está nos dois subconjuntos. Seria isso uma regra? Não. A semântica presente no universo por nós observado é que deverá nos levar a definir essa exclusividade ou não. Observe que no exemplo dado para ANIMAIS temos subconjuntos de ANIMAIS DOMÉSTICOS e ANIMAIS SELVAGENS e, dependendo do caso, poderemos ter um mesmo animal (uma onça de estimação) estando presente nos dois subconjuntos simultaneamente. Isso nos leva a explorar em nossos modelos dois tipos distintos de especializações:

- Especializações mutuamente exclusivas.
- Especializações não mutuamente exclusivas.

Especializações Mutuamente Exclusivas (Categorias)

O enquadramento de uma estrutura de Gen-Espec como mutuamente exclusiva se dará quando, a partir da observação e identificação dos elementos e subconjuntos existentes em um conjunto, percebermos que, sempre que um elemento do conjunto global pertencer a um dos subconjuntos, não poderá pertencer, simultaneamente, a outro.

Perceba que, pela definição dada, não excluímos a possibilidade de um elemento mudar, ou migrar, de um dos subconjuntos para o outro. A única restrição imposta é a participação em dois, ou mais, subconjuntos simultaneamente. Assim, se partimos de um conjunto com 10 elementos, o somatório dos elementos pertencentes a cada um dos subconjuntos deverá ser obrigatoriamente 10.

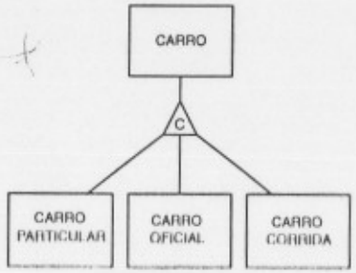


Vamos imaginar uma situação onde tenhamos observado e abstraído um conjunto global chamado CARRO. Nesse conjunto, para fins de especialização, podemos ter definido os subconjuntos: CARRO PARTICULAR, CARRO OFICIAL e CARRO DE CORRIDA. Pela análise da situação existente, e pelas regras impostas pelo próprio ambiente observado, temos que um CARRO PARTICULAR não pode ser usado para fins oficiais ou para competições. Um CARRO OFICIAL também não pode ser utilizado em competições ou usado para fins particulares (pelo menos neste exemplo!). E, finalmente, um CARRO DE CORRIDA não pode ser usado para fins oficiais ou particulares.

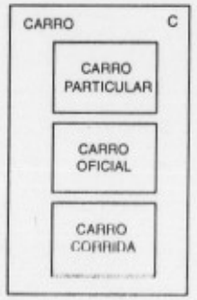
Do ponto de vista de modelagem, podemos caracterizar a existência desses subconjuntos mutuamente exclusivos como uma estrutura onde são enquadrados elementos em "categorias" distintas.

Essa visão de categorias ajuda-nos a reforçar o fato de que se um elemento pertence a uma categoria não pertencerá a outra. Essa atribuição do conceito "categoria" é simplesmente arbitrada para diferenciação entre um caso e outro. Poderíamos estar utilizando qualquer outra denominação desde que deixássemos claro a mútua exclusividade. Assim, teremos o conjunto CARRO englobando três categorias distintas: os oficiais, os particulares e os de corrida.

Alguns autores não denotam essa exclusividade no modelo gráfico; entretanto, achamos importante e útil essa representação, pois, conceitualmente, esse é um fato bastante relevante. Uma das notações possíveis para essa situação é:



notação 1



notação 2

Utiliza-se uma notação escolhida para a estrutura de Gen-Espec, seja ela qual for, e acrescenta-se junto ao elemento separador dos subconjuntos a identificação de que os subconjuntos são mutuamente exclusivos. No nosso exemplo, utilizamos a letra "C" junto ao triângulo formador dos subconjuntos e junto ao conjunto global para denotar a existência de CATEGORIAS. Qualquer outra representação que trouxesse a mesma semântica poderia ser utilizada.

Neste exemplo, segundo a própria definição, não permitimos que um CARRO pertença simultaneamente a duas categorias. Entretanto, não excluímos a possibilidade de um CARRO PARTICULAR vir a se transformar em um CARRO OFICIAL. O fato é que se isso ocorrer ele deixará de ser um CARRO PARTICULAR e figurará somente no novo subconjunto.

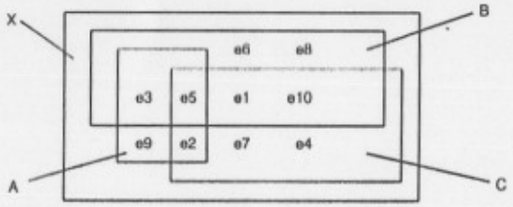
Especializações Não-mutuamente Exclusivas (Papéis)

A ocorrência de especializações não-mutuamente exclusivas também poderá ser encontrada durante o processo de modelagem de um ambiente observado. Caracterizar, especificamente, se um elemento pertence a uma especialização com ou sem exclusividade deverá ser o objetivo do processo de investigação das regras estabelecidas pelo ambiente. Algumas vezes essa própria regra poderá vir a ser alterada, com o passar do tempo, fazendo com que uma especialização mutuamente exclusiva se transforme em não-mutuamente exclusiva e vice-versa.

Para que possamos estabelecer equivalência entre um caso e outro estabeleceremos, também, o conceito de uma especialização não-mutuamente exclusiva.

> *Dado um elemento pertencente a um conjunto global, diz-se que ele pertence a uma especialização não-mutuamente exclusiva se este puder pertencer a um ou mais subconjuntos do conjunto global simultaneamente.*

Graficamente, poderíamos perceber esse tipo de ocorrência através do estabelecimento de um conjunto X e da formação de três subconjuntos: A, B, e C.



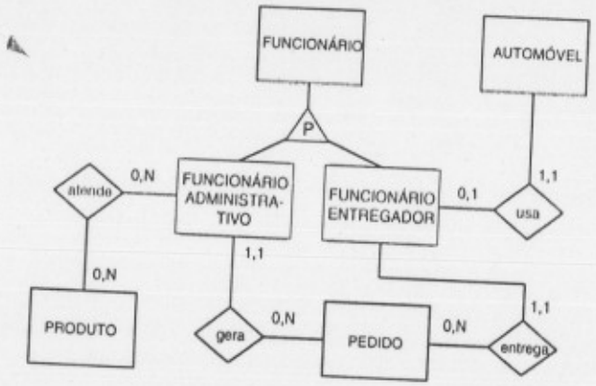
Perceba que só temos um elemento e2, mas que ele pode pertencer simultaneamente aos conjuntos A e C. Assim, também o elemento e3 pertence aos conjuntos B e A simultaneamente. É esse tipo de ocorrência que pretendemos modelar através de uma estrutura de especialização não-mutuamente exclusiva. Nessa situação, apesar de possuirmos apenas 10 elementos no conjunto X, o somatório dos elementos pertencentes a cada um dos subconjuntos será maior que 10, exatamente 16 (A=4, B=6 e C=6).

Para fins de exemplificação, vamos imaginar um modelo sendo desenvolvido em um ambiente empresarial e, nesse ambiente, sendo identificados os seguintes objetos:

- FUNCIONÁRIOS
- AUTOMÓVEIS
- PRODUTOS
- AGENDAMENTO DE ENTREGAS
- PEDIDOS

Dentre os FUNCIONÁRIOS, pode ser percebida a existência de subgrupos distintos caracterizados como FUNCIONÁRIOS ENTREGADORES e FUNCIONÁRIOS ADMINISTRATIVOS. Os que pertencem ao subgrupo dos FUNCIONÁRIOS ENTREGADORES estarão alocados a AGENDAMENTO DE ENTREGAS de determinados PEDIDOS. Já os FUNCIONÁRIOS ADMINISTRATIVOS serão responsáveis pelo gerenciamento e atendimento de vendas de determinados PRODUTOS. Acontece que, durante épocas de alto volume de vendas (Natal, promoções, etc.), os FUNCIONÁRIOS ADMINISTRATIVOS muitas vezes são alocados para efetuar entregas, bem como FUNCIONÁRIOS ENTREGADORES são recrutados para atendimento de vendas.

Com essa situação, pode-se perceber que, em um dado instante, um certo FUNCIONÁRIO poderá ser visto simultaneamente como FUNCIONÁRIO ENTREGADOR e FUNCIONÁRIO ADMINISTRATIVO. Isso caracterizará o que chamamos de PAPEL: a possibilidade de um elemento "atuar" como dois personagens diferentes em um mesmo ambiente, tal como um artista de novela que interpreta dois personagens ao mesmo tempo.



Perceba que um FUNCIONÁRIO ADMINISTRATIVO possui relacionamento com PRODUTO (atende), enquanto que um FUNCIONÁRIO ENTREGADOR possui relacionamento com PEDIDO (entrega) e com AUTOMÓVEL (usa). Se analisássemos o funcionário João Geraldo observando-o, em um dado instante, como FUNCIONÁRIO ENTREGADOR, poderíamos identificar o AUTOMÓVEL utilizado por ele para fazer as entregas dos PEDIDOS a ele destinados.

Não haveria sentido, neste instante, desejar obter informações sobre os PRODUTOS que têm atendimento pelo funcionário João Geraldo. Assim também se, em um outro instante, observássemos o funcionário João Geraldo como FUNCIONÁRIO ADMINISTRATIVO, poderíamos identificar os PRODUTOS aos quais ele atende e não poderíamos querer identificar o AUTOMÓVEL a ele destinado.

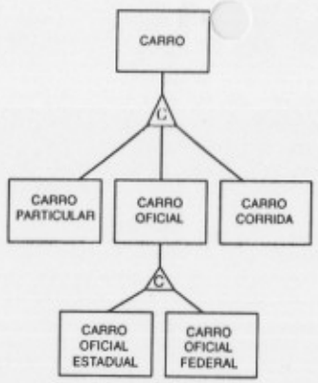
Quando tentamos manter a visão de "papéis" simultâneos mas, ao mesmo tempo, desvinculados, estamos abordando o aspecto puramente conceitual atribuído ao modelo em questão. Muitos devem estar, já neste instante, preocupados em como esse tipo de modelo será implementado sob a forma de tabelas relacionais. Lembre-se de que ainda não estamos preocupados com isso! Veremos, ao devido tempo, as regras para a derivação desse tipo de estrutura conceitual para as estruturas lógicas (modelos relacionais) necessárias. Isso também se aplica aos modelos com especializações mutuamente exclusivas. Se você pensou em subconjuntos distintos como tabelas separadas para a implementação, ainda não atingiu o grau de abstração conceitual desejado. Procure esquecer, por enquanto, o aspecto de implementação e volte-se para o aspecto conceitual. O que desejamos é, simplesmente, modelar grupos e suas subdivisões, alocando aos subconjuntos os atributos e relacionamentos específicos a eles pertencentes, identificando sua mútua exclusividade ou não.

Níveis de Especialização

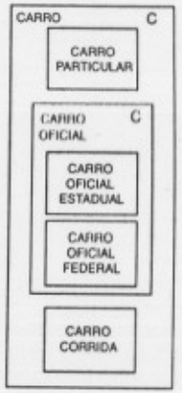
Outra característica que podemos encontrar, quanto à formação de subconjuntos especializados, é a da existência de diferentes níveis de agrupamento. Nesse caso, partindo-se de um conjunto global, poderemos estabelecer um primeiro nível de subconjuntos e, dentro desses, por sua vez, voltar a estabelecer novos subconjuntos.

Vamos utilizar o mesmo exemplo relativo aos CARROS para demonstrar essa situação. Já vimos que o conjunto CARRO pode ser dividido em subconjuntos. Caracterizamos esses subconjuntos como CARROS PARTICULARES, CARROS OFICIAIS e CARROS DE CORRIDA e estabelecemos o critério de exclusividade. Agora, imagine que tenhamos também observado que os CARROS OFICIAIS podem ser subdivididos em CARROS OFICIAIS ESTADUAIS e CARROS OFICIAIS FEDERAIS. Essa subdivisão não invalida a anterior, mas sim complementa-a. Por sua vez, apesar de válida ela só se aplica aos CARROS OFICIAIS e, portanto, não pode ser vinculada ao conjunto CARRO.

Logo, para representar esse tipo de ambiente observado deveríamos construir o modelo:



notação 1



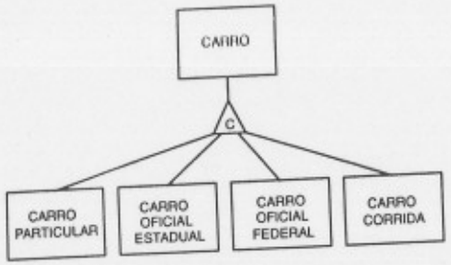
notação 2

A hierarquização entre estruturas de especialização demonstrará assim que, a partir de um grupo mais generalizado, obtivemos um nível de especialização inicial e que, a partir deste, obtivemos uma nova especialização. Em princípio, esse tipo de estrutura não tem limite para níveis de hierarquização, mas, do ponto de vista prático, dois níveis têm demonstrado ser o limite suficiente para o tratamento da quase totalidade dos casos.

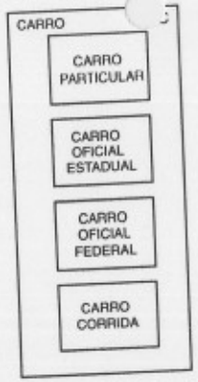
Uma característica que deve ser observada na expansão dos níveis verticais de uma estrutura de especialização é que, não necessariamente, todos os níveis expandidos deverão ter o mesmo tipo de definição dos subconjuntos. Isso significa que no primeiro nível poderemos criar subconjuntos baseados em "categorias", e que no segundo nível poderemos, se assim for o ambiente observado, definir a separação dos subconjuntos baseados em "papéis". No exemplo dado, para fins de facilitação do entendimento, utilizamos sempre a separação dos subconjuntos baseados em "categorias".

Caso não se deseje utilizar mais de um nível de hierarquização, pode-se utilizar o artifício de expandir horizontalmente a estrutura ao invés de expandi-la verticalmente. Conceitualmente, acabaremos por ter demonstrar uma mesma situação apesar de que, em alguns casos, poderemos estar tendo de alocar atributos de modo repetitivo em locais distintos.

Vejamos um exemplo dessa situação. Aproveitando o modelo já descrito onde criamos dois níveis de hierarquização, vamos representar o caso dos CARROS OFICIAIS, CARROS PARTICULARES e CARROS DE CORRIDA através de uma estrutura com um só nível de hierarquia.

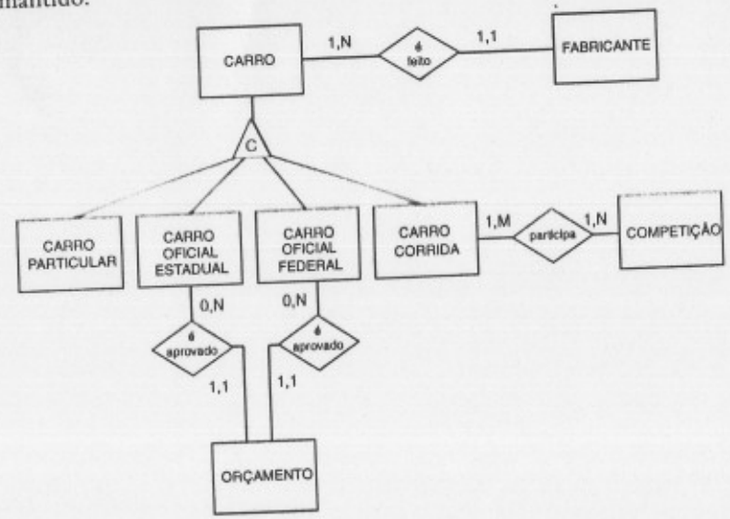


notação 1



notação 2

Perceba que, nesse caso, acabamos por expandir horizontalmente nossa estrutura de árvore (aumentamos a amplitude e não a profundidade). Isso pode parecer um elemento facilitador do modelo mas nem sempre será a solução ideal. O que aconteceria se tivéssemos relacionamentos que devessem ser estabelecidos com a especialização CARROS OFICIAIS? E como seriam alocados atributos específicos de CARROS OFICIAIS? A resposta é: repetindo-os nas duas estruturas. Isso acabaria por tornar nosso modelo, apesar de semanticamente correto, mais difícil de ser representado e mantido.



- **Atributos:**

CARRO	CARRO OFICIAL ESTADUAL
cor	horário de início de disponibilidade
modelo	horário final de disponibilidade
potência	finalidade
número de cilindros	secretaria de estado à qual serve

CARRO PARTICULAR	CARRO OFICIAL FEDERAL
placa	horário de início de disponibilidade
ano de licenciamento	horário final de disponibilidade
valor acumulado de multas	finalidade
	ministério do governo ao qual serve

CARRO DE CORRIDA
modelo do motor
modelo do chassi
equipe a que pertence
valor de patrocínio

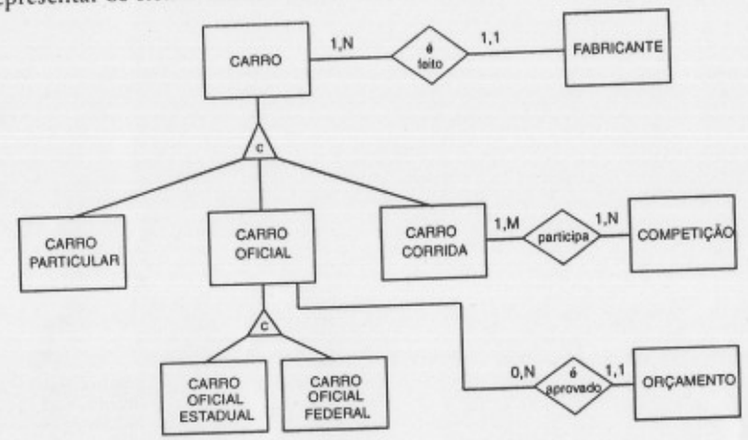
Caso estivéssemos trabalhando com um nível de especialização mais detalhado, utilizando-se de uma estrutura com dois níveis, poderíamos ter um modelo mais facilmente representável. Tanto a alocação dos atributos como dos relacionamentos seria feita unicamente na entidade CARROS OFICIAIS e, seguindo o princípio de herança, seriam repassados para as entidades "filhas".

A característica de herança é um dos mecanismos que torna a aplicação das estruturas de Generalização-Especialização um meio de facilitação da construção de modelos. Através de sua utilização podemos fazer com que simplificações sejam introduzidas na representação dos objetos por nós observados.

Sempre que uma característica, seja ela um atributo ou um relacionamento, for comum a todos os elementos dos subgrupos existentes para um dado grupo, esta característica poderá ser "fatorada" ou "colocada em evidência" junto à entidade que representa a generalização. Desse modo, por herança, os subconjuntos herdarão essas características. Por outro lado, todas as características que sejam específicas para um subconjunto deverão ser especializadas junto a esse subconjunto, para que não sejam repassadas nem ao grupo generalizador, nem aos demais subgrupos.

Características (atributos e relacionamentos) que sejam comuns a todos os elementos devem ser alocadas na entidade generalizadora, e características que sejam específicas de um subgrupo devem ser alocadas a ele.

Usando esse princípio nosso modelo anteriormente apresentado para representar os elementos de CARROS acabaria por se transformar em:



- Atributos:

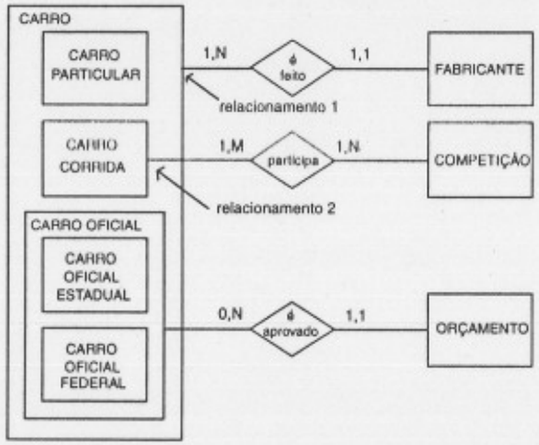
CARRO	CARRO OFICIAL
cor	horário de início de disponibilidade
modelo	horário final de disponibilidade
potência	finalidade
número de cilindros	

CARRO PARTICULAR	CARRO OFICIAL ESTADUAL
placa	secretaria de estado à qual serve
ano de licenciamento	
valor acumulado de multas	

CARRO DE CORRIDA	CARRO OFICIAL FEDERAL
modelo do motor	ministério do governo ao qual serve
modelo do chassi	
equipe a que pertence	
valor de patrocínio	

Perceba que os atributos comuns aos dois subgrupos de CARROS OFICIAIS agora só aparecem uma única vez junto ao grupo que representa a generalização. Isso já vinha ocorrendo antes, também, para os atributos de CARRO (cor, modelo, potência e número de cilindros) que por serem aplicáveis a todos os tipos de carros haviam sido mencionados uma única vez na entidade generalizadora. Por herança, esses atributos (e relacionamentos) são transferidos "para baixo" passando a ser também aplicáveis às entidades especializadas.

Outra característica a ser observada é que os relacionamentos que chegam à entidade generalizada se aplicam a todos os elementos independentemente do conjunto. Os relacionamentos que chegam a uma especialização se aplicam a apenas aqueles elementos ali contidos. Na representação utilizada anteriormente, isso fica bastante claro; entretanto, quando se utiliza um tipo de representação alternativa onde os subtipos ficam dentro da entidade supertipo, deve-se tomar o cuidado de representar claramente os relacionamentos. Aqueles que se aplicam à generalização devem "tocar" a entidade generalizadora e parar aí (ver relacionamento 1). Aqueles que se aplicam a uma especialização devem passar pela entidade generalizadora e "tocar" internamente a entidade especializada (ver relacionamento 2).

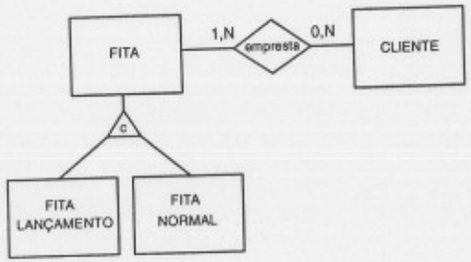


Quando Utilizar a Estrutura de Generalização-Especialização

Temos visto muitas vezes a estrutura de Generalização-Especialização sendo utilizada de modo ineficaz como simples meio para exemplificar os domínios, ou tipos de elementos, existentes em um ambiente. Outras vezes, tenta-se através dela estabelecer regras de negócio que pouco têm a ver com o inter-relacionamento entre elementos do modelo. Apesar de conceitualmente não implicar um erro, acaba-se por "poluir" o modelo com a inclusão

de elementos que efetivamente não traduzem qualquer caracterí[illegible] semântica diferencial.

Imagine o caso onde durante a elaboração de um modelo de uma videolocadora foram observados dois subgrupos distintos de fitas de vídeo: as FITAS DE LANÇAMENTO e as FITAS NORMAIS. Por se verificar que esses subgrupos faziam parte de um grupo maior chamado FITAS poderia se ter concebido um modelo tal como:



- Atributos:

FITA	FITA DE LANÇAMENTO
número da fita	sem atributos específicos
título	
data de aquisição	
produtora	

CLIENTE	FITA NORMAL
número do cliente	sem atributos específicos
nome do cliente	
data de nascimento	
endereço	

Neste caso, apesar de tornar explícito que o conjunto FITAS está dividido em dois grandes subgrupos, não identificamos nenhuma característica, de atributos ou relacionamentos, específica para os subgrupos especializados. Caso nosso modelo permitisse (por razões de tamanho e destinação), poderíamos até manter essa representação, mas, certamente, isso não ocorrerá em grande parte dos casos práticos e reais.

Assim como regras práticas para definir se uma especialização deve ou não ser explicitada no modelo, devemos avaliar os seguintes pontos:

- **Regra 1.** Existe [illegible]m atributo que seja aplicável a somente um subconjunto de elementos e não a todos? Se sim, então é desejável a criação da especialização.
- **Regra 2.** Existe algum relacionamento que seja aplicável a somente um subconjunto de elementos e não a todos? Se sim, então é desejável a criação da especialização.
- **Regra 3.** Não estaremos "poluindo" o modelo com a inserção de detalhes desnecessários? Não, não estaremos. Então, é possível a criação da especialização.
- **Regra 4.** A finalidade de nosso modelo é procurar incorporar o máximo de elementos gráficos para que a compreensão não dependa de análise de elementos complementares, tais como dicionários de dados, documentos, anexos? Sim, é. Então é possível a criação da especialização.

Neste ponto voltamos a lembrar algo que já discutimos com mais detalhes em tópicos anteriores:

> *Incluir semântica em um modelo não significa incluir excesso de informação, mas sim a informação necessária e suficiente para a compreensão exata dos conceitos a serem transmitidos.*

Essas regras nos levam a ter duas situações distintas:

1. Situação em que a especialização *deve ser* explicitada: Nesse caso enquadramos as duas primeiras regras vistas anteriormente. São situações em que, além de possível, é fortemente recomendada a aplicação da especialização. Essa recomendação se baseia tanto na fidelidade ao modelo a ser gerado como na adequação de futuras estruturas de dados a serem implementadas. Caso não se utilize a especialização estaremos mascarando características particulares aplicáveis a um ou outro elemento e então criando futuros pontos de instabilidade no modelo.

2. Situação em que a especialização *pode ser* explicitada. Aqui aplicam-se as duas últimas regras definidas anteriormente. São situações em que, apesar de não obrigatória, é possível a utilização de uma estrutura de especialização. A definição de quando aplicar ou não esse recurso dependerá de bom senso e prática. Muitas vezes poderemos estar sobrecarregando o modelo com informações desnecessárias e, outras vezes, tornando-o, simplesmente, adequado ao uso a que se propõe. É difícil estabelecer uma regra infalível para o julgamento do uso ou não da especialização nesses casos. O único critério indiscutivelmente aceito é: use o bom senso!

Como Apresentar os Subgrupos Existentes

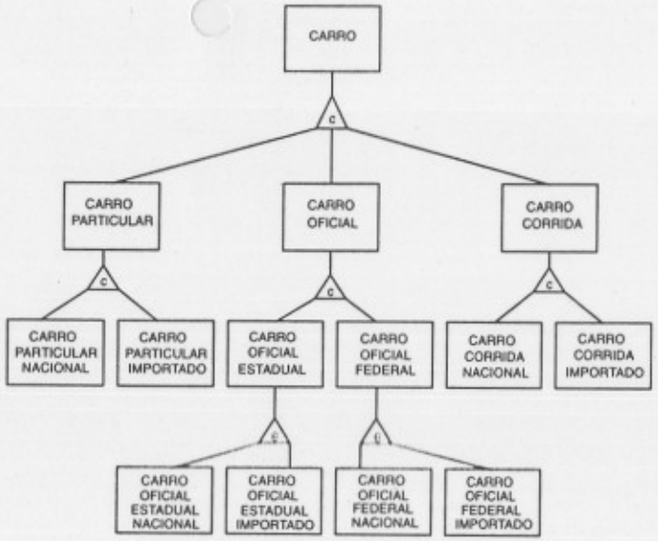
Dentro dos tópicos já vistos até agora, pudemos explorar uma das principais características da estrutura de Generalização-Especialização: a hierarquização de subgrupos. Essa hierarquia, como vimos, pode ser em um, dois ou mais níveis dependendo da complexidade e das características que queiramos modelar para um ambiente.

Entretanto, outra habilidade também está disponível para ser utilizada junto à estrutura de Generalização-Especialização: a definição de critérios. Um critério é um modo, ou enfoque, de julgamento para a classificação dos elementos de um conjunto. A partir desse critério teremos, então, a formação dos subgrupos pela divisão dos elementos pertencentes ao conjunto global.

Isso poderia parecer já uma habilidade explorada, pois já tratamos desse modo nossos conjuntos nos exemplos vistos até agora. Todavia, algo que ainda não tratamos é a possível construção simultânea de mais de uma estrutura de Generalização-Especialização sobre o mesmo conjunto. Nesse caso, partindo-se de um conjunto global inicial, poderemos representar tantas subdivisões por critérios diferentes, quantas queiramos.

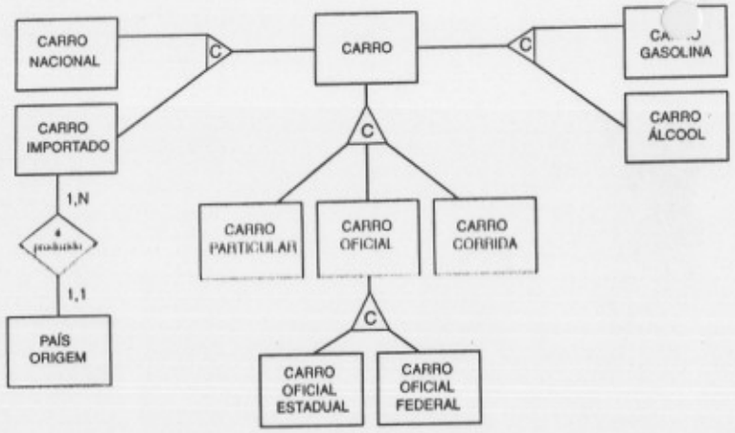
Vamos continuar a explorar o exemplo dos CARROS. Vamos partir do ponto já atingido, onde temos uma estrutura de Generalização-Especialização de dois níveis sob o critério "finalidade de uso do carro" (particular, oficial, corrida). Vamos estender nosso enfoque de avaliação do conjunto de CARROS e verificar o critério "nacionalidade do carro". Podemos verificar a existência de CARROS classificados em: CARROS IMPORTADOS e CARROS NACIONAIS.

Perceba que esse critério não desabilita o anterior e vice-versa. Um CARRO OFICIAL pode ser um CARRO NACIONAL ou um CARRO IMPORTADO. Um CARRO PARTICULAR também, e um CARRO DE COMPETIÇÃO, se pensarmos em categorias que utilizam carros de passeio (como a Fórmula Turismo) também. Algumas pessoas até poderiam pensar em utilizar uma estrutura que combinasse todas essas situações, onde então teríamos algo como:



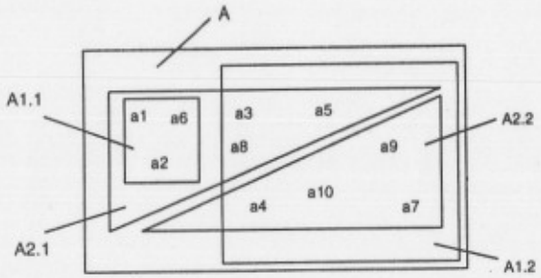
Nota-se, entretanto, que esse tipo de estrutura, gerada pela composição (análise combinatória) dos diversos critérios, não nos leva sempre a uma situação adequada para a modelagem. Imagine ainda que alguns relacionamentos e atributos devam ser aplicados, assim como já foram para o critério de "finalidade de uso do carro", para o novo critério "nacionalidade do carro". Isso implicaria propagar as repetições de relacionamentos e atributos para cada uma das especializações onde aparecessem respectivamente os tipos NACIONAL e IMPORTADO. Não vamos procurar exemplificar esse tipo de situação, pois seria impraticável. Imagine se mais um terceiro critério de separação fosse acrescentado para classificar também os CARROS em função do seu tipo de combustível (álcool ou gasolina). Teríamos um modelo de proporções ainda maiores, pois teríamos que novamente dividir cada uma das especializações já existentes em duas novas especializações.

Para resolver esse problema, vamos propor a utilização do recurso de *critérios simultâneos de especialização*. Veja qual seria o resultado para representarmos a situação apresentada mesmo com o critério adicional de "tipo de combustível" e com um relacionamento estabelecido para os CARROS IMPORTADOS.



No exemplo dado, acabamos por utilizar três diferentes critérios de separação de subconjuntos, mas todos baseados em "categorias" e nenhum em "papéis". Isso não é necessariamente uma regra. Esse recurso foi utilizado somente para facilitar o entendimento. Poderíamos ter imaginado o caso de um CARRO que pudesse utilizar como combustível tanto álcool como gasolina. Nesse caso, teríamos no critério "tipo de combustível" uma criação de subgrupos assumindo "papéis" e não "categorias", o que seria extremamente correto.

Do ponto de vista semântico, a criação de diferentes "critérios de especialização" para um mesmo conjunto poderia ser representada graficamente como:



Perceba que para o conjunto A, composto de 10 elementos (a1 até a10), estabelecemos dois critérios de especialização. O primeiro representado por delimitações quadradas criou as categorias A1.1 e A1.2 compostas respectivamente pelos elementos (a1, a2, a6) e (a3, a5, a8, a4, a10, a9, a7). Paralelamente estabelecemos um segundo critério, representado

por delimitações triangulares, que criou as categorias A2.1 e A2.2 compostas respectivamente pelos elementos (a1, a6, a3, a5, a2, a8) e (a4, a10, a9, a7). Cada um dos critérios compõe diferentes divisões para o conjunto global A, devendo, portanto, serem analisados desvinculadamente. Entretanto, deve-se sempre levar em consideração que um mesmo elemento que apareça em um dos subconjuntos sob um critério também aparecerá em outro sob outro critério.

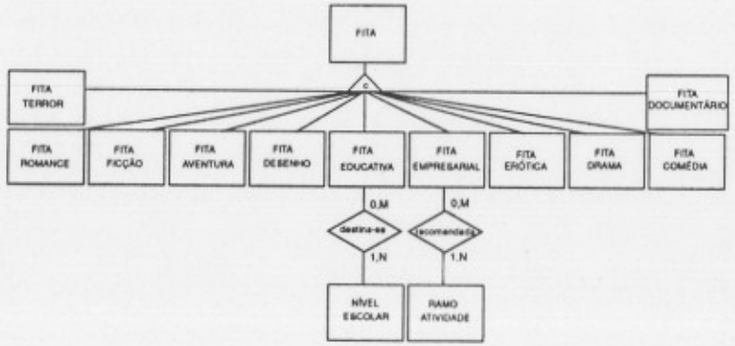
Outro ponto importante a ser observado na definição das estruturas de especialização e em suas representações é a abrangência dos subconjuntos a serem tratados. Esse ponto, muitas vezes, pode nos levar a situações complexas, pois o universo de tipos de elementos existentes pode ser bastante grande.

Vamos imaginar um caso onde para fins de explicitação estivéssemos criando uma estrutura de especialização a partir da entidade FITA em um ambiente de videolocadora. O critério utilizado seria "classificação da fita" e teríamos definido que este se apresenta sob a modalidade de "categoria" e não de "papel".

Dentre as classificações poderíamos encontrar, por exemplo: romance, ficção, aventura, desenho, educativa, empresarial, erótica, drama, comédia, terror e documentário.

Para fins de nossa representação teríamos necessidade, nesse exemplo, de só agregar alguns poucos relacionamentos e atributos especializados nas fitas do tipo empresarial e educativa. No primeiro tipo alocaríamos o ramo de atividade da empresa ao qual se destina a fita. No segundo, o nível escolar ao qual se destina a fita.

Imagine que se partíssemos para uma representação padrão da estrutura de especialização teríamos algo como:

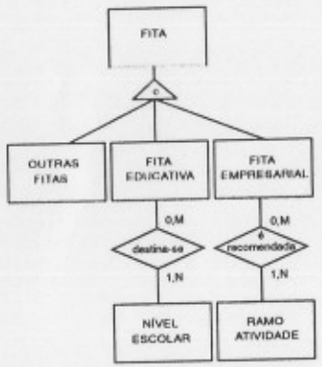


Pode-se notar que, apesar de semanticamente correta, essa representação é bastante extensa e, em meio a um modelo completo, tornar-se-ia inviável ou, pelo menos, improdutiva. Entretanto, como a condição que

estabelecemos e representamos demonstra, somente para os tipos de FITA EDUCATIVA e FITA EMPRESARIAL existem detalhes que nos interessam explicitar. Então, por que não representar somente esses dois tipos de FITA na especialização? Resposta: Porque assim acabaríamos por induzir a interpretação de nosso modelo ao entendimento de que na videolocadora só existem esses dois tipos de fitas.

Para superar esse obstáculo propomos, portanto, que seja utilizado um artifício de representação que acabará por manter a correta semântica do modelo e que, ao mesmo tempo, reduzirá suas proporções. Esse artifício consiste na criação de uma especialização que represente todas as especialidades que não nos interesse representar. Essa especialização pode receber o nome simbólico de OUTRAS, OUTROS, DEMAIS, ou aquele que parecer melhor aplicável.

Para que fique mais claro, utilizaremos esse recurso em nosso exemplo. Veja qual seria o resultado para representar o mesmo ambiente observado através da aplicação desse recurso:



Perceba que acabamos por denotar no modelo somente os subconjuntos onde havia interesse de caracterização de atributos ou relacionamentos especializados. Apesar disso, não omitimos o fato de que dentre os possíveis subtipos de FITAS existem outras que não as FITAS EDUCATIVAS e FITAS EMPRESARIAIS. Para todas as OUTRAS FITAS poderão, eventualmente, até ser estabelecidos atributos e relacionamentos específicos. O fato importante é que para todas aquelas que não as FITAS EDUCATIVAS e FITAS EMPRESARIAIS existem características comuns que ficam representadas pelo subconjunto OUTRAS FITAS.

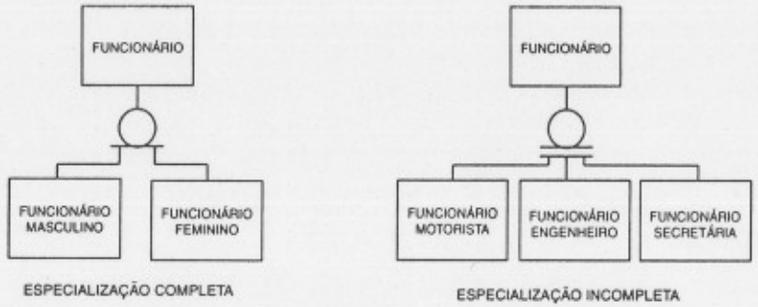
Esse é um recurso que não encontramos formalmente definido em outros autores mas que, na prática, tem demonstrado ser bastante útil durante o processo de modelagem conceitual. Mesmo na derivação de modelos lógicos para implementação (em bases de dados relacionais) esse artifício continua sendo aplicável e válido.

Há de se ressaltar que para o completo entendimento do que representa a especialização OUTRAS FITAS deverá ser elaborado um adequado dicionário de dados no qual sejam elencadas as possíveis "classificações de fita" por ela representadas.

Exemplo:

OUTRAS FITAS: representa o subconjunto de todas as fitas disponíveis para locação na videolocadora e que pertençam às classificações: romance, ficção, aventura, desenho, erótica, drama, comédia, terror e documentário.

Algumas notações, tais como a usada na metodologia IDEF1X, denotam esse caso como uma especialização incompleta. Isso significa que, dentre os elementos representados graficamente, não estão incluídos todos os tipos de elementos existentes, mas somente alguns deles. Em contraposição, no caso de especializações completas, pode-se, através da simbologia utilizada, concluir que os tipos enumerados são a totalidade dos tipos existentes.



Utilizando a Estrutura de Gen-Espec em Auto-relacionamentos

A utilização da estrutura de Generalização-Especialização pode ser um recurso bastante interessante para acrescentar semântica aos casos de auto-relacionamento. Como já vimos, um auto-relacionamento se dá quando diferentes instâncias de um mesmo tipo de objeto se relacionam entre si. Nesses casos, apesar de um relacionamento recursivo resolver satisfatoriamente nossa necessidade de mapeamento, acabamos por ter um modelo onde pouco se pode garantir sobre qual elemento se relaciona efetivamente com outro.

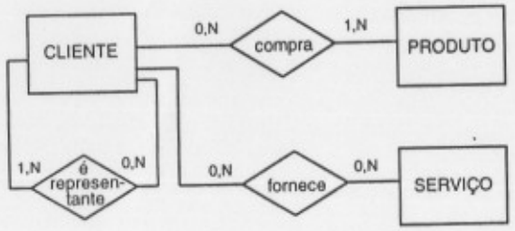
Nesse tipo de relacionamento, normalmente existe o estabelecimento de dois papéis, ou categorias, com os quais se estabelecerão as ligações. Nessa situação é importante que se possa explicitar quais papéis, ou categorias, são estes e como eles se relacionam. Somente através de um re-

lacionamento recursivo não teremos satisfeitas as condições para ex[illegible]itar essas regras de relacionamento. Assim, agregando-se a estrutura de Generalização-Especialização poderemos atingir a semântica necessária.

Exemplo:

Suponha que estejamos modelando o ambiente da empresa AutoPeças Ltda., produtora de autopeças, que, para redução de custos, adquire do mercado externo SERVIÇOS de terceiros (outras empresas) para fabricação e acabamento de seus produtos. A AutoPeças Ltda., por sua finalidade básica, comercializa para vários CLIENTES seus PRODUTOS.
Acontece que dentre os CLIENTES, compradores de seus PRODUTOS, existem PESSOAS FÍSICAS e PESSOAS JURÍDICAS. As PESSOAS FÍSICAS são, em alguns casos, representantes comerciais de algumas PESSOAS JURÍDICAS (empresas). Em alguns casos, essas PESSOAS JURÍDICAS são as mesmas que acabam por fornecer SERVIÇOS à AutoPeças Ltda.

Nosso modelo básico, sem o uso da estrutura de Generalização-Especialização, acabaria por ser representado como algo assim:

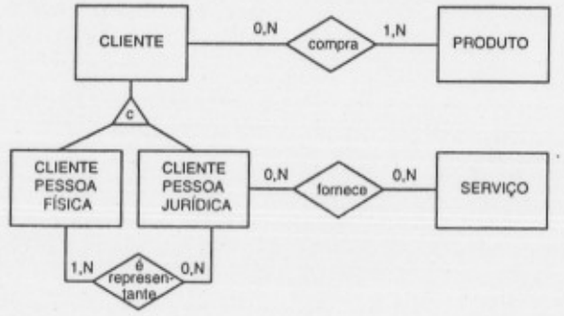


Perceba que, apesar de correto, temos poucos elementos para interpretar todas as informações necessárias. Como poderíamos saber, por exemplo, pelo modelo, se os clientes se dividem em dois tipos distintos de CLIENTES?

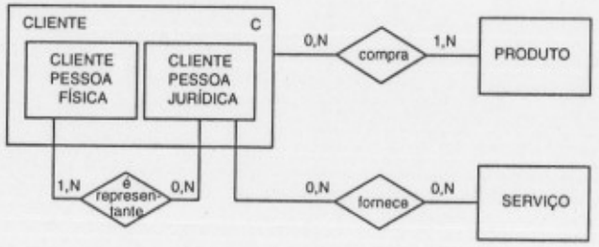
Um cliente PESSOA FÍSICA pode representar outro PESSOA FÍSICA
Um cliente PESSOA JURÍDICA pode representar outro PESSOA JURÍDICA
Um cliente PESSOA JURÍDICA pode representar uma PESSOA FÍSICA
Um cliente PESSOA FÍSICA pode fornecer SERVIÇOS
Um cliente PESSOA JURÍDICA pode fornecer SERVIÇOS

Por não podermos [illegible]rpretar claramente essas situações no modelo, é que consideramos este modelo como "pouco semântico". Para aumentar sua semântica, iremos agora utilizar a estrutura de Generalização-Especialização e representar o mesmo ambiente observado. Perceba a mudança:

A) NOTAÇÃO 1



B) NOTAÇÃO 2



Utilizamos dois modos diferentes de representação da especialização somente para mostrar como um ou outro padrão de representação se aplicaria a esse caso.

Estrutura de Agregação

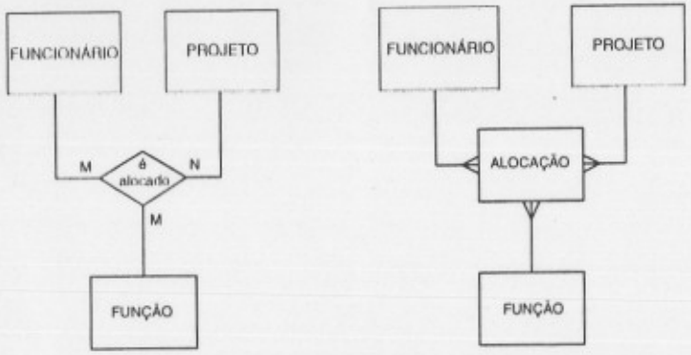
A estrutura de agregação é outro recurso que poderá ser aplicado aos modelos para facilitar o entendimento semântico e tornar mais claros os graus dos relacionamentos ternários, ou de maior número.

Quando apresentamos o item que discutia o número de elementos que participa de um relacionamento, vimos que poderíamos ter um, dois, três, quatro ou mais tipos de objetos envolvidos. A existência de somente um tipo de objeto implicava o estabelecimento de auto-relacionamentos entre diferentes instâncias desse objeto. Quando tínhamos dois tipos de objetos, estávamos trabalhando com o tipo de situação padrão, normalmente encontrada. Quando tínhamos três ou mais elementos, podíamos

representar também, utilizando os recursos gráficos, esse tipo de relacionamento quer na notação de Martin ou de Chen. Vimos que na notação de James Martin havia a necessidade de mudança de representação gráfica de relacionamentos ternários e que na representação de Peter Chen isso não era um fator obrigatório apesar de possível.

Acontece que se optarmos pela notação de Chen para a representação de relacionamentos ternários, ou de maior número, acabaremos por ter o estabelecimento de um relacionamento onde a interpretação do "grau do relacionamento" ou "cardinalidade" é bastante prejudicada. Suponha o caso onde um FUNCIONÁRIO pode exercer diversas FUNÇÕES e uma FUNÇÃO possa ser exercida por diversos FUNCIONÁRIOS. Suponha também que a definição da FUNÇÃO exercida pelo FUNCIONÁRIO seja feita em cada um dos PROJETOS onde esse FUNCIONÁRIO esteja alocado. O FUNCIONÁRIO poderá estar alocado a diversos PROJETOS, mas em cada um dos PROJETOS cada FUNCIONÁRIO só pode exercer uma FUNÇÃO. Isso não impede, entretanto, que um FUNCIONÁRIO acabe por estar alocado a diversas FUNÇÕES em diferentes PROJETOS.

Vejamos qual seria a representação básica para esse ambiente observado sem a utilização da agregação:



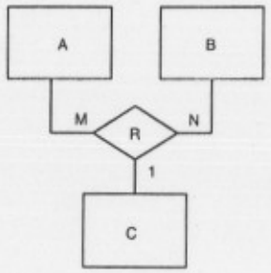
Representação segundo Peter Chen

Representação segundo James Martin

Perceba que apesar de termos, em ambos os casos, a representação de que uma alocação a um PROJETO é de um FUNCIONÁRIO em uma FUNÇÃO, não conseguimos estabelecer claramente a regra:

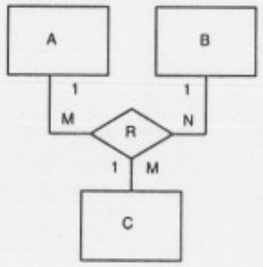
> *Um FUNCIONÁRIO pode exercer várias FUNÇÕES em diversos PROJETOS mas somente uma em um dado PROJETO.*

Isso faz com que, apesar de correto, nosso modelo fique "pouco semântico", e em alguns casos até confuso. No caso anterior, por exemplo, temos que o relacionamento entre FUNCIONÁRIO e FUNÇÃO é de MxM. Ou seria entre PROJETO e FUNÇÃO NxM, ou entre FUNÇÃO e FUNCIONÁRIO NxM??? Note que os graus dos relacionamentos acabam por se misturar de certo modo onde não podemos mais estabelecer qual deles predomina sobre os demais. Nesse caso, temos todos os graus iguais a MxN, mas e se um deles fosse 1:N? Como interpretar?



Seria obrigatório que, se o relacionamento entre C e A fosse de grau 1xN, assim também fosse o relacionamento entre C e B, pois se entre C e B tivéssemos um relacionamento de MxN teríamos que substituir a indicação de 1 existente por M, voltando ao problema inicial, pois prejudicamos agora o relacionamento entre C e A.

Ainda mais, mesmo que o relacionamento entre C e A fosse de 1xN e o de C e B também fosse de 1xM, não quer dizer que obrigatoriamente o relacionamento de A para B fosse de MxN. E se ele fosse de 1x1? Para todos esses casos, deveríamos prever algo como:



Mesmo assim, qual diríamos, agora, ser o grau do relacionamento R?

1x1x1?
1xMxN?
Mx1x1?
ou outro qualquer ...

Espero que tenha ficado claro que nesse caso não há como apresentar todas as informações necessárias utilizando uma estrutura padrão. Vamos ver agora a resolução desse impasse utilizando a agregação.

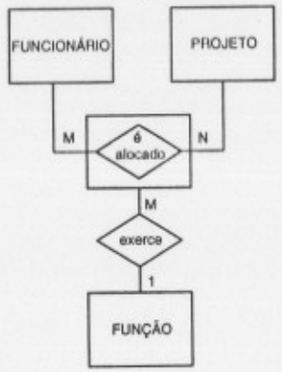
Por definição, a agregação é:

> *Uma estrutura que denota a existência de uma junção de elementos através de um relacionamento, e que permite que essa junção seja percebida como um novo elemento a ser, por sua vez, relacionado a outro elemento.*

Isso faz com que seja possível a representação de um conjunto de elementos ternários, ou de maior número, pela associação gradual de dois elementos a um terceiro, e assim por diante. Como já vimos anteriormente, não é correto o desmembramento de uma relação ternária em duas relações binárias simples. Com isso estaremos introduzindo falhas no processo associativo. Entretanto, o que propomos agora é o desmembramento de uma relação ternária em uma relação binária, que por sua vez se associe a um novo elemento.

Desse modo, podemos manter a integridade entre as associações não gerando combinações inválidas. Para maiores detalhes, veja o exemplo dado quando é apresentado o conceito de relações ternárias.

Do ponto de vista de nosso exemplo, com relação a FUNCIONÁRIOS, PROJETOS e FUNÇÕES, poderíamos, aplicando esse novo recurso, gerar um modelo como se segue:



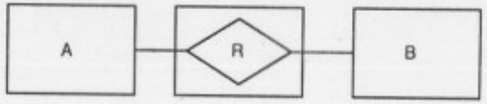
Nesse modelo, pode-se inclusive modelar o caso onde um FUNCIONÁRIO pode exercer várias FUNÇÕES em vários PROJETOS mas somente uma em um determinado PROJETO.

Perceba que a AGREGAÇÃO, graficamente falando, é um misto de ENTIDADE e RELACIONAMENTO. Sua própria notação gráfica procura demonstrar que a ENTIDADE que participa do relacionamento "exerce" é obtida pelo RELACIONAMENTO entre PROJETO e FUNCIONÁRIO. Assim, cada par da associação PROJETO x FUNCIONÁRIO pode estar ligado a uma e só uma FUNÇÃO, denotando que um FUNCIONÁRIO, apesar de poder participar de diversos PROJETOS, quando estiver alocado a um deles estará somente em uma FUNÇÃO.

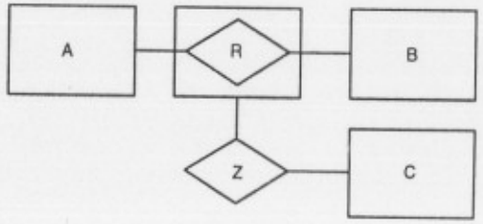
A correta representação dessa estrutura, até para que seu entendimento seja mais fácil, deverá tomar como base:

- O estabelecimento inicial de um RELACIONAMENTO "R" entre duas entidades A e B:

- A transformação do relacionamento "R" em uma nova ENTIDADE através de um símbolo que assim o represente:

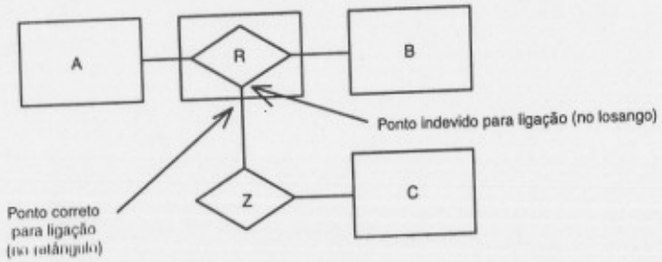


- O estabelecimento de um novo RELACIONAMENTO (Z) entre a ENTIDADE criada (R) e a terceira entidade envolvida (C):



Note que todos os relacionamentos agora existentes (R e Z) se processam entre duas entidades. R é a associação de A e B. Z é a associação de AB e C. Perceba também que o relacionamento Z se estabelece com a entidade (RETÂNGULO) existente entre A e B. Algumas pessoas, ao

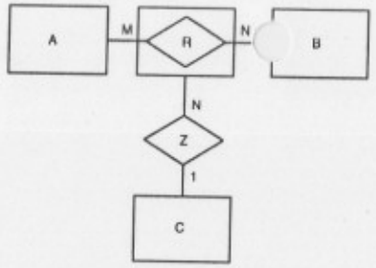
modelar, muitas vezes ligam o relacionamento Z ao relacionamento R. Isso acaba por confundir a futura leitura do próprio modelo; portanto, tome esse cuidado e seu modelo ficará muito mais fácil de ser compreendido.



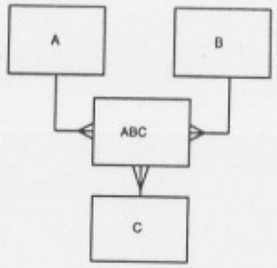
Muitas pessoas, ao verem uma estrutura dessa já concebida, têm dificuldades em interpretá-la. Para facilitar a interpretação, pense em como ela foi formada. Lembre-se dos três passos que foram executados para criá-la:

1. Foi estabelecido um relacionamento entre duas entidades (procure as duas entidades e um relacionamento unindo-as).
2. Foi transformado esse relacionamento em entidade (identifique um retângulo delimitando um relacionamento entre duas entidades).
3. Foi estabelecido um novo relacionamento com o terceiro elemento (identifique a terceira entidade conectada ao relacionamento).

Muitas pessoas acabam por não utilizar esse tipo de recurso no modelo conceitual, optando pelo uso de uma entidade associativa, ou uma representação do tipo James Martin (que força o uso de entidades associativas para relacionamentos ternários). Grande parcela destes, que optam por outros recursos, acredita que um modelo conceitual "maior", ou com mais entidades, levará forçosamente a um número maior de tabelas durante a fase de implementação lógica. Isso é falso! Veremos mais adiante que as regras de derivação de uma estrutura de agregação nos levarão de volta a um modelo "enxuto" e otimizado.



modelo com o uso da agregação



modelo com o uso da entidade associativa

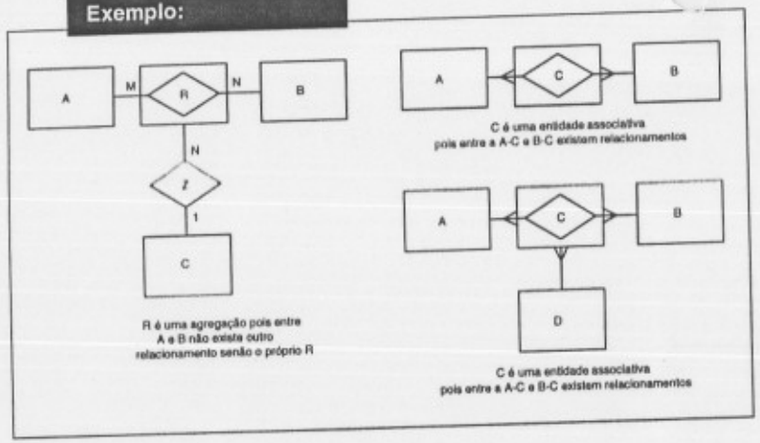
Não se preocupe com o modo como essa estrutura irá ser implementada quando você partir para o projeto lógico. Acredite, por enquanto, que:

- Não haverá proliferação de tabelas.
- A estrutura final obtida com o uso da agregação será idêntica àquela que seria obtida se tivéssemos utilizado uma entidade associativa.

Um ponto importante a ressaltar é que alguns autores utilizam uma notação específica para representar uma entidade associativa (aquela que representa um relacionamento MxN sem ou, principalmente, com atributos). Essa notação confunde-se com a mesma que utilizamos para denominar uma agregação. Isso poderá causar algumas dúvidas em um primeiro instante mas, com a experiência, poderemos identificar cada caso sem risco de interpretação equivocada.

Caso a representação seja sempre utilizada para substituir um relacionamento MxN, e entre ela e as demais entidades existam relacionamentos 1:N, então estaremos nos defrontando com uma entidade associativa. Caso o grau de relacionamento não seja, obrigatoriamente, MxN e entre as entidades formadoras da notação não existam outros relacionamentos senão aquele existente na própria notação, então teremos uma agregação. Outro fator a ser avaliado é a presença obrigatória de um relacionamento estabelecido com o elemento analisado. Se esse relacionamento nem sempre existir, estaremos tratando de uma entidade associativa. Já se o relacionamento existir, poderemos estar tanto tratando de um caso de agregação como de um caso de entidade associativa. Nessa situação, deve-se analisar as cardinalidades e os relacionamentos como já descrevemos anteriormente.

Exemplo:



Notações Gráficas Utilizadas na Modelagem Conceitual

A seguir, apresentaremos de modo sintetizado algumas das notações de representação utilizadas para a modelagem conceitual (e lógica em alguns casos). Algumas delas têm maior disseminação no mercado, tais como a de Peter Chen e a de James Martin (já utilizadas até aqui em diversos exemplos); entretanto, todas cumprem, ao seu modo, a finalidade a que se propõem. Devemos estar preparados, durante nossa atuação na área de modelagem, para nos defrontarmos com diferentes notações em diferentes metodologias. O importante será sermos capazes de identificar quais são os elementos semânticos de cada uma delas e aplicá-los corretamente. Após isso, teremos assimilado uma nova "linguagem" para expressar ou receber os conceitos que tenhamos captado em um determinado ambiente observado.

Notação IDEF1X

Essa metodologia, apesar de poder ser aplicada no nível conceitual, incorpora em suas raízes muitas características de projeto de banco de dados (modelagem lógica, portanto). Sua diagramação é bastante dependente de conceitos tais como chaves e dependências de identificação. Isso faz com que, sob o ponto de vista prático, seja muito mais indicada para o projeto lógico de bancos de dados (relacionais principalmente) do que para outros níveis de abstração. Dentre os elementos que essa notação abrange estão:

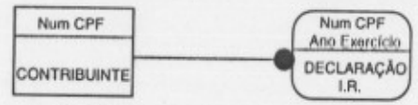
Entidades

As entidades podem ser enquadradas em diferentes categorias:

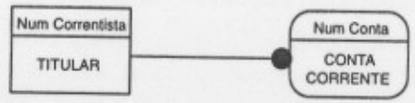
- Independentes ou dependentes.
- Associativas ou não-associativas.
- Dependentes de identificador e/ou dependentes de existência.
- Subtipos ou supertipos.

Uma entidade independente, representada por um retângulo, é aquela que possui existência e identificador (chave) próprio, isto é, não composto por qualquer outro identificador de outra entidade. Já uma entidade dependente, representada por um retângulo com os cantos arredondados, pode ser dependente de identificador ou dependente de existência. A dependência de identificador significa que uma entidade necessita, para formar sua própria chave identificadora, da chave de outra entidade. Já a dependência de existência implica que, para que uma ocorrência de uma entidade filha exista, deverá existir, relacionada a ela, obrigatoriamente, uma ocorrência da entidade pai.

entidades independentes

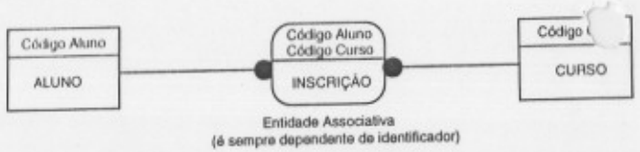


Declaração I.R. é dependente de chave e de existência

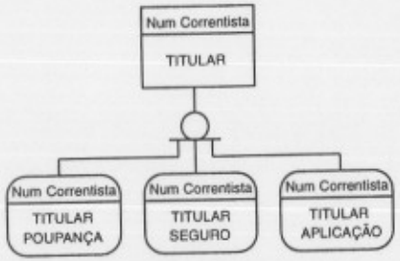


Conta Corrente é dependente de existência

Uma entidade associativa, por definição, é dependente de identificador de suas entidades pai, portanto, é representada como uma entidade dependente qualquer. Entidades subtipo também são representadas como entidades dependentes, já que têm dependência de existência.



Entidade Associativa
(é sempre dependente de identificador)



Entidade Subtipo
(é sempre dependente de existência)

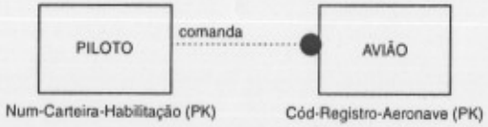
Relacionamentos

As notações utilizadas para relacionamento indicam simultaneamente quatro aspectos:

- Cardinalidade.
- Opcionalidade de existência (dependência de existência).
- Dependência de identificador.
- Significado (role).

Assim sendo, ao interpretarmos um relacionamento podemos identificar, também, além das características semânticas básicas, características de implementação tais como migração de chaves estrangeiras e independência de linhas em tabelas. Isso, mais uma vez, denota a orientação dessa metodologia para o modelo lógico. Perceba que no modelo conceitual, em princípio, não temos necessidade de conhecer as chaves identificadoras de cada uma das entidades. Portanto, em um modelo conceitual, não é necessário que a notação gráfica denote essas características.

Sempre que um relacionamento for implementado entre duas entidades onde haja dependência de identificador (a chave da entidade filha utiliza atributos da entidade pai para sua formação), teremos uma linha contínua. Caso contrário (a entidade filha não é dependente de identificador), teremos uma linha tracejada.



Após estabelecido o critério de dependência de identificador, pode-se agregar ao modelo a cardinalidade do relacionamento utilizando o seguinte padrão:

- Para denotar "zero, um ou muitos" (0,N) utiliza-se somente uma bola fechada

significado : "um avião realiza zero, um ou vários vôos"

- Para denotar "um ou muitos" (1,N) utiliza-se uma bola fechada e a letra "P"

significado : "um avião possui pelo menos 1 tripulante e até vários deles"

- Para denotar "zero ou um" (0,1) utiliza-se uma bola fechada e a letra "Z"

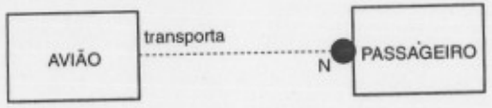
significado : "um avião necessita de nenhum ou de um só engenheiro de vôo"

- Para denotar "obrigatoriamente muitos", utiliza-se uma bola fechada com a letra "N"

significado : "um avião transporta necessariamente diversos passageiros"

Finalmente, para indicar a possibilidade de existência de uma ocorrência da entidade filha sem a existência da equivalente ocorrência na entidade pai (independência de existência ou opcionalidade de relacionamento), utiliza-se um losango junto à linha que sai da entidade pai.

significado : "um avião atende necessariamente a diversos passageiros e um passageiro pode ser atendido por nenhum ou um avião"



significado : "um avião transporta necessariamente diversos passageiros e um passageiro deve ser transportado por um e sempre um avião"

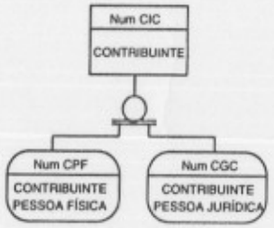
Atributos

Não existe uma notação gráfica específica para denotar os atributos e suas funções (chaves primárias, chaves estrangeiras, índices, etc.). O que tem sido visto junto às ferramentas CASE que implementam essa técnica é a alocação dos atributos "dentro" da entidade. Caso eles sejam chaves primárias, são seguidos da indicação PK e, caso sejam chaves estrangeiras, são seguidos da indicação FK. Essa foi a notação que utilizamos em diversos exemplos durante a fase de derivação dos modelos lógicos apresentados.

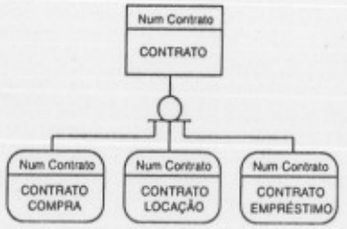
Mais uma vez lembramos que a identificação de chaves primárias e chaves estrangeiras é um forte indício de uma técnica voltada ao processo de modelagem lógica e não de modelagem conceitual.

Subtipos e Supertipos

Para a representação de estruturas de generalização-especialização são utilizados dois símbolos gráficos distintos. O primeiro identifica uma explosão incompleta da entidade, ou seja, existem alguns subtipos não mostrados na explosão. O segundo símbolo identifica uma explosão completa onde todos os possíveis subtipos existentes estão representados após a explosão.



EXPLOSÃO COMPLETA
(não existe outro tipo de contribuinte possível)

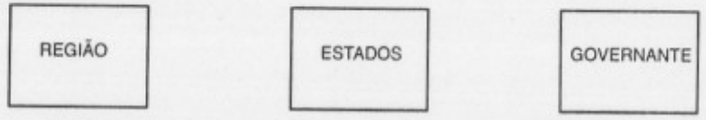


EXPLOSÃO INCOMPLETA
(poderia haver outros tipos de contrato, p. ex., LEASING)

Notação de Bachman

A notação hoje atribuída a Bachman (a notação de setas, que veremos mais adiante) é, na verdade, uma derivação da diagramação original proposta por Charles Bachman para a representação de estruturas de dados. Em sua notação básica existiam apenas dois elementos: as estruturas de dados (registros) e os relacionamentos existentes entre elas.

Para representar as entidades era utilizado um retângulo. Não havia conceitos de entidade fraca ou entidade forte, dependência de identificadores, chaves primárias, etc.

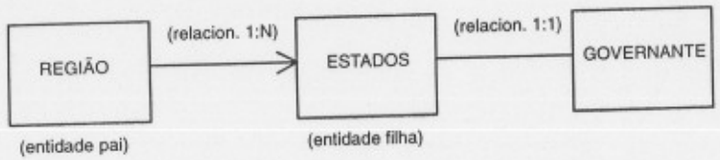


Para identificar os tipos de associatividade entre as estruturas de dados eram usados apenas dois tipos de representação:

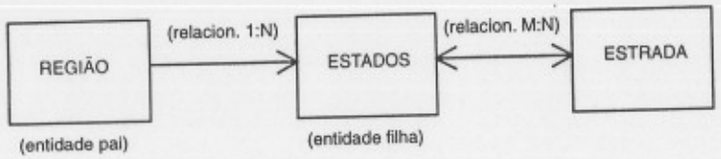
- A cardinalidade 1 era representada por uma linha sem terminação.
- A cardinalidade N era representada por uma linha com uma ponta de seta.

Não eram agregados elementos para indicar opcionalidade nas ligações ou para denotar graus mínimos e máximos. A dependência entre

ocorrências dos registros se estabelecia automaticamente sempre entidade pai (aquela em que o grau da cardinalidade era 1) para a entidade filha (aquela em que o grau da cardinalidade era N).



Caso houvesse um relacionamento de grau M:N haveria, simplesmente, uma linha com duas pontas de setas, uma em cada sentido.



A representação de atributos também não era feita no modelo gráfico, ficando somente a cargo de especificações adicionais feitas em meio descritivo convencional. Não era dada ênfase a maiores características semânticas e, portanto, elementos tais como estruturas de generalização-especialização ou agregação não existiam.

Atualmente, algumas ferramentas CASE, assim como para outras notações, implementam recursos gráficos e semânticos para representação de elementos tais como entidades fracas e fortes, entidades associativas, opcionalidade de relacionamentos, subtipos, etc. Esses recursos são chamados de *extensões* das notações originais. Eles podem ou não estar disponíveis em uma ou outra ferramenta específica e, se presentes, poderão disponibilizar mais semântica aos modelos gerados.

Exemplo:

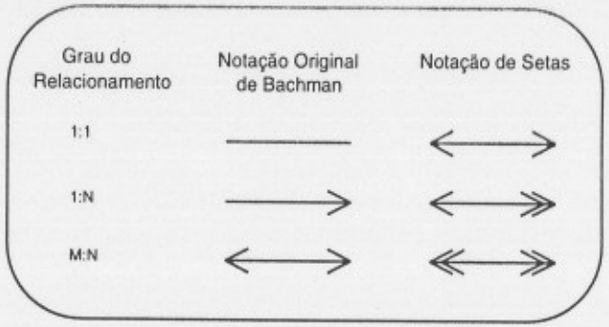
Alguns elementos adicionais encontrados na metodologia de Bachman presente em uma das ferramentas CASE disponível no mercado (na verdade, a notação referenciada como Bachman nessa ferramenta é a notação de setas que veremos no próximo tópico).

Notação de Setas

Essa notação é derivada da notação apresentada no tópico anterior e, muitas vezes, tem sido denominada também de *notação de Bachman*. Não encontramos na literatura alguma referência que atribua essa derivação ao próprio Bachman nem tampouco a qualquer outro autor. Alguns autores e ferramentas CASE utilizam a denominação *notação de setas* como uma denominação formal. Portanto, utilizaremos aqui, também, este termo para diferenciar essa representação gráfica da notação original de Bachman.

O tratamento dado à representação das entidades e dos atributos é bastante próximo de todas as demais notações voltadas a modelos lógicos. São apresentados os atributos "dentro" das entidades, acompanhados de indicações de chaves primárias ou estrangeiras. A identificação dos critérios de dependência de existência ou dependência de identificador não é comumente feita, apesar de algumas ferramentas CASE, por já terem esse recurso implementado para outras notações, aplicá-lo também à notação de setas.

O grande diferencial dessa notação está na representação gráfica da cardinalidade dos relacionamentos. A indicação de graus mínimos e máximos também não será comumente vista e ficará a cargo da implementação de uma ou outra ferramenta CASE. A evolução na representação original se deu da maneira:



Também nessa notação, apesar de se tratar de uma evolução da notação original de Bachman, não existem, em princípio, elementos de caracterização semântica adicional (estruturas de generalização-especialização e agregação). Continuam, entretanto, válidas as observações já feitas com relação a recursos específicos possivelmente existentes em uma ou outra ferramenta CASE.

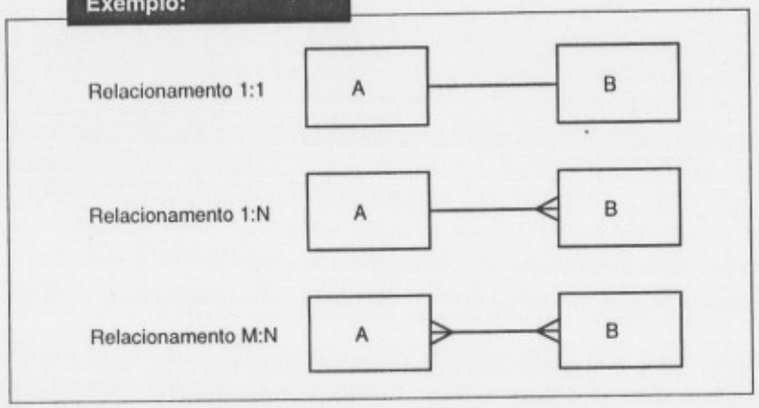
Notação "Pé-de-Galinha" (*Crow's foot*)

A notação conhecida como *pé-de-galinha*, ou no original *Crow's foot* (pata de corvo), não abrange, na verdade, todo um conjunto de elementos gráficos para representação de modelos de dados. Ela somente trata, em especial, da notação utilizada para a representação dos relacionamentos. James Martin incorporou essa notação junto a outros elementos e disseminou, através de seus trabalhos sobre Engenharia da Informação, essa notação que hoje muitas vezes é tida como a notação de Martin. Na verdade, a notação de Martin (que veremos a seguir) é muito mais completa, pois incorpora a notação de pé-de-galinha e dá-lhe elementos de semântica adicional.

Em seu modo mais elementar, a notação pé-de-galinha utiliza somente duas simbologias gráficas para denotar o grau de um relacionamento:

- A cardinalidade 1 é representada por uma linha sem terminação.
- A cardinalidade N é representada por uma linha com um pé-de-galinha.

Exemplo:



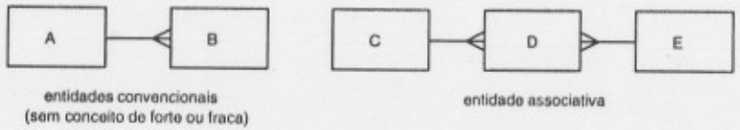
Notação de Martin (Engenharia da Informação)

A notação proposta por James Martin vem sendo atualizada, com o passar do tempo, através de seus diversos trabalhos já publicados. A cada nova edição, novos meios de representação vêm sendo apresentados.

Martin, um dos precursores e disseminadores da Engenharia da Informação, sempre deu grande ênfase à necessidade e ao papel dos modelos de dados. Assim, praticamente desde o início, já víamos em seus modelos elementos de semântica adicional. Caracterizações de graus mínimos e máximos de relacionamentos, opcionalidades, subtipos, contingência e exclusividade de relacionamentos e outros elementos já estavam presentes em sua proposta inicial. Não havia muita ênfase a critérios de dependências de identificador, chaves estrangeiras, chaves primárias e outros elementos voltados ao modelo lógico, o que demonstra uma boa aplicabilidade dessa notação para os modelos conceituais. O único ponto a ser considerado como mais ligado ao modelo lógico do que ao modelo conceitual é o fato de que relacionamentos com atributos têm de ser mapeados através de entidades associativas obrigatoriamente, o que, de certa maneira, desvirtua a semântica conceitual. Apesar disso, com certeza, esta tem sido uma das notações mais utilizadas nos últimos tempos.

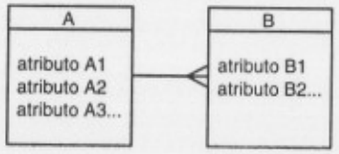
Entidades

O tratamento de entidades é bastante simples, limitando-se a uma única representação. Não há distinção entre entidades fortes e entidades fracas nem representação de modo diferenciado para as entidades associativas. Somente um retângulo é utilizado para denominar graficamente qualquer tipo de entidade:



Atributos

Os atributos não são representados no modelo gráfico, ficando listados à parte. Em algumas ferramentas CASE, os atributos podem aparecer enumerados dentro do retângulo representativo da entidade:

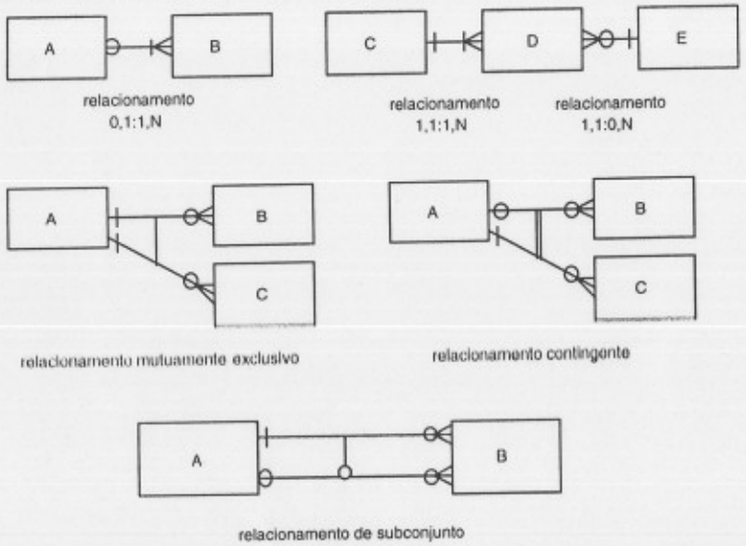


Esse recurso pode aparecer em outras notações. Quase sempre essa característica está associada ao editor gráfico da ferramenta CASE. Se ele permite alocar os atributos para uma das notações que usa, permitirá também para as demais.

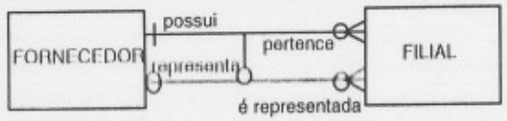
Relacionamentos

A notação básica utilizada para relacionamentos é a de pé-de-galinha. A ela foram adicionados elementos de caracterização do grau mínimo do

relacionamento (relacionamentos condicionais), elementos para definição de contingência, exclusividade e subconjuntos.

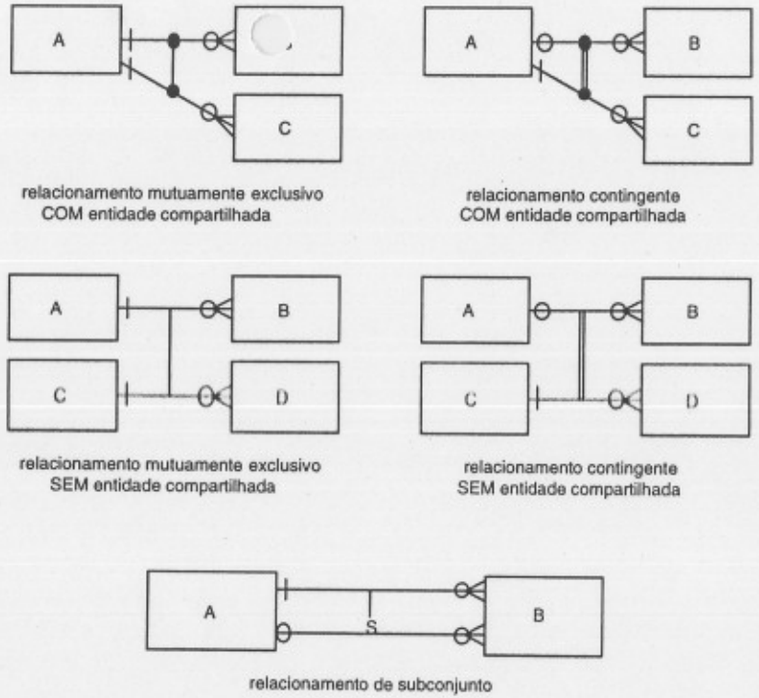


Os conceitos de relacionamentos exclusivos e contingentes já foram apresentados anteriormente, mas o conceito de relacionamentos de subconjunto é novo. Esse tipo de relacionamento ocorre quando entre duas entidades podem ser estabelecidos dois ou mais relacionamentos e entre esses relacionamentos existe uma associação de subconjunto. Vejamos um exemplo:

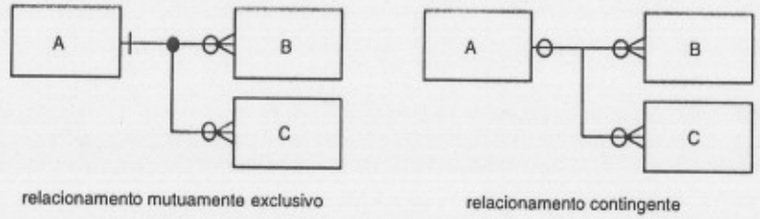


Nesse caso, podemos perceber que entre FORNECEDOR e FILIAL existem dois relacionamentos distintos. O primeiro indica todas as filiais que um fornecedor possui. O segundo, que é um subconjunto do primeiro, indica quais filiais são representadas pelo próprio fornecedor. O sentido de subconjunto fica assim estabelecido: "de um grupo de filiais pertencentes a um fornecedor podemos obter um subconjunto de filiais representadas por ele".

Em alguns trabalhos, Martin apresenta alguns padrões diferentes de notação. Passa a ser adotada uma representação diferenciada para relacionamentos contingentes e mutuamente exclusivos entre entidades compartilhadas e não compartilhadas. Também a representação de relacionamentos de subconjuntos muda de um círculo para uma letra "s".



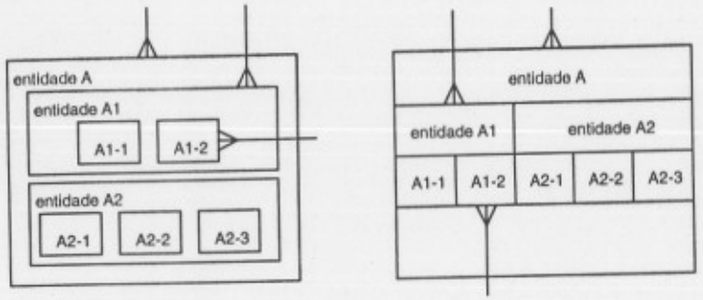
Também podem ser encontradas outras representações para relacionamentos mutuamente exclusivos e relacionamentos contingentes em trabalhos apresentados por Martin. No livro *Técnicas estruturadas e CASE*[2] é apresentada a seguinte notação:



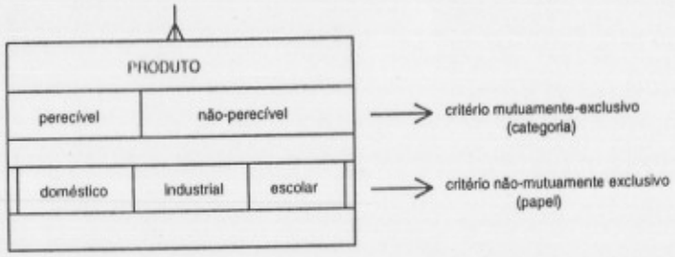
Como se pode perceber, várias notações podem ser aplicadas para representar um mesmo "conceito", inclusive por um mesmo autor. Isso demonstra a importância do conhecimento do significado de cada representação e da disseminação desse significado para os envolvidos no processo de modelagem.

Estrutura de Generalização-Especialização

Em seus primeiros trabalhos sobre Engenharia da Informação, James Martin não apresentava nenhuma notação específica para elementos semânticos desse tipo. Com o passar do tempo, foram sendo introduzidas representações específicas para generalização-especialização. Perceba no exemplo a seguir duas notações diferentes para representar uma mesma estrutura de especialização em dois níveis.



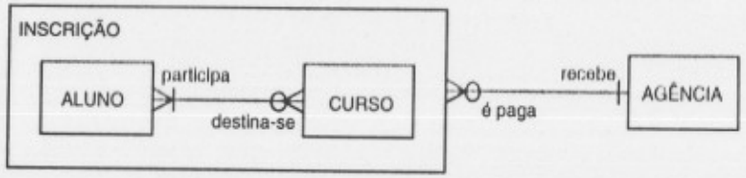
Adicionalmente, na segunda notação apresentada na figura, podemos caracterizar se a especialização é do tipo categoria ou papel, ou seja, se é mutuamente exclusiva ou não. Isso é feito através de um símbolo colocado às margens da subdivisão dos elementos. Veja o exemplo:



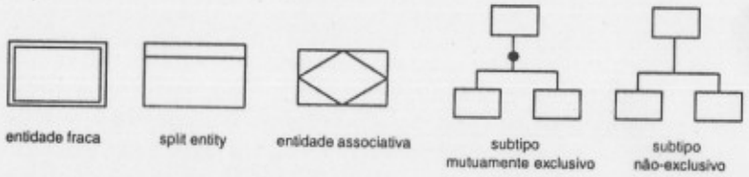
Estrutura de Agregação

Apesar de não se poder atribuir exatamente a Martin a representação que mostraremos a seguir para estruturas de agregação, iremos incluí-la nesse item pois é comum se encontrar esse tipo de diagramação em modelos que aplicam o padrão de Martin. É um tipo de representação que utiliza uma estrutura similar ao primeiro tipo de notação mostrado para estruturas de generalização-especialização (entidades desenhadas "dentro" de outras entidades).

Define-se primeiram[illegible] o relacionamento básico entre duas entidades e, após isso, representa-se uma entidade "envolvendo" todo esse grupo para que então ele, como uma "agregação de dados", possa ser relacionado com as demais entidades necessárias.



Ainda alguns outros elementos gráficos poderão ser encontrados em ferramentas CASE disponíveis no mercado associados à notação de Martin. Entre eles temos:



Dentre esses elementos gráficos, a representação de entidade associativa é a mais freqüentemente encontrada, apesar de não pertencer, originalmente, à notação da Engenharia da Informação. As demais, apesar de aplicáveis, enquadram-se nas facilidades adicionais disponibilizadas por uma ou outra ferramenta CASE, como já mencionamos anteriormente.

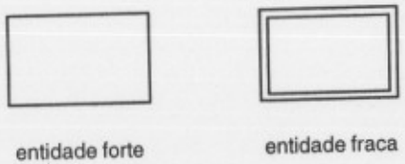
Notação de Peter Chen

Em seu trabalho original, Peter Chen apresentava um conjunto de elementos voltados somente à representação de entidades, atributos e relacionamentos. Não existiam elementos adicionais tais como estruturas de generalização-especialização ou agregações. Por outro lado, bastante ênfase era dada aos aspectos de dependência de existência e dependência de identificador, pois, apesar de voltada à representação da estrutura inerente dos dados, a visão de Chen vislumbrava uma estreita ligação com o projeto de estruturas físicas de implementação desses modelos, fossem as estruturas relacionais ou não.

Entidades

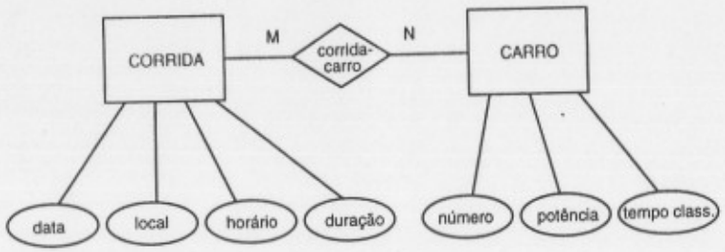
Chen foi o responsável pela introdução dos conceitos de "entidades fortes" e "entidades fracas". Como já vimos esses conceitos anteriormente,

vamos somente representar aqui cada uma das notações utilizada[illegible]or outro lado, o conceito de "entidade associativa" não era tratado por Chen, já que um relacionamento de grau MxN com atributos (se necessário) era representado, ainda, como um relacionamento, sem a necessidade de criação de uma "entidade associativa". Assim sendo, as representações possíveis para entidades eram:

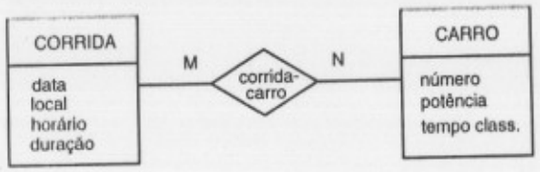


Atributos

A notação proposta originalmente por Chen para a representação dos atributos pertencentes a uma entidade era pouco prática, pois eram utilizados balões para representar cada um dos atributos. Isso fazia com que entidades com cinco ou mais atributos já tivessem alguma dificuldade de representação.

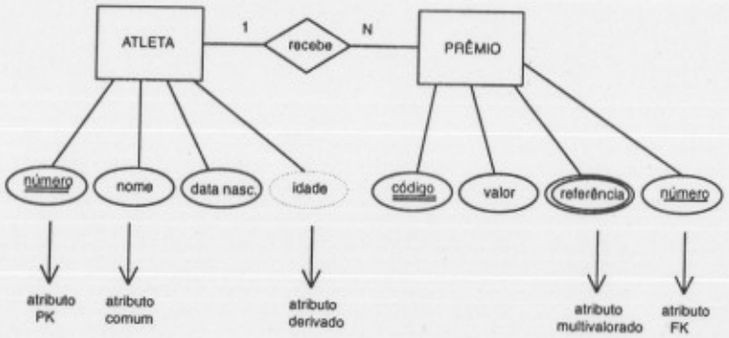


Hoje, mesmo a ferramenta CASE comercializada pela empresa de Chen não mais utiliza essa notação, adotando o método de alocação dos atributos "dentro" da entidade.



Algumas ferramentas CASE ainda permitem a utilização da notação original de Chen, possibilitando, adicionalmente, a caracterização do tipo de atributo. Tem-se, então, um tipo de balão diferente para atributos comuns, outro para atributos das chaves primárias, outro para atributos que

sejam chaves estrangei[illegible] outro para atributos que sejam multivalorados e outro para atributos que sejam obtidos por derivação (cálculo ou processamento) em tempo de acesso:



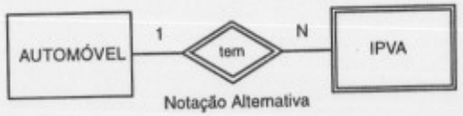
Relacionamentos

Chen foi o introdutor da utilização de um losango (ou *diamond*) como elemento gráfico para a representação de um relacionamento. Atualmente, uma "nova" metodologia para modelagem orientada a objetos (o método FUSION) passou a incorporar também essa notação para associações (conceito equivalente a um relacionamento, como já vimos). Isso demonstra que, apesar de pouco difundida e, às vezes, até considerada em "desuso", como cita o manual de uma ferramenta CASE bastante difundida no mercado, essa notação é bastante adequada para a representação de objetos e seus relacionamentos.

Não há diferenciação na representação gráfica de relacionamentos de grau 1:1, 1:N ou M:N. A única variação, em termos de representação gráfica, se dá quando representamos um relacionamento que estabelece vínculo de dependência de existência ou dependência de identificador. No trabalho original de Chen, eram utilizadas letras específicas para denotar uma e outra dependência. A letra "E" denominava dependência de existência e as letras "ID" denotavam dependência de identificador. Uma seta indicava a direção da dependência e a entidade dependente era sempre uma entidade fraca.

Atualmente, tem-se adotado outro padrão de representação para denotar dependência. Esse novo padrão foi, basicamente, resultado da necessidade de se utilizar um outro tipo de indicação de dependência que não um texto inserido dentro do losango:

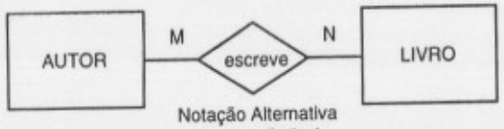
Notação Original



Notação Alternativa

Outra diferença que pode ser observada na notação de relacionamentos diz respeito ao texto existente dentro do losango. Originalmente, Chen utilizava nesse espaço somente um texto representativo das entidades que participavam do relacionamento ou as letras E e ID (como vimos anteriormente). Com o passar do tempo e com a tentativa de agregar maior semântica aos modelos criados, passou-se a adotar o padrão de representação do significado ou da finalidade do relacionamento no espaço existente dentro do losango. Como se trata de um texto e não de um símbolo, essa característica passou a ser tratada indiferentemente pelas ferramentas CASE existentes no mercado. A escolha do conteúdo desse texto passou a ser um critério pessoal ou a depender do grau de semântica desejado no modelo:

Notação Original
(sem semântica)

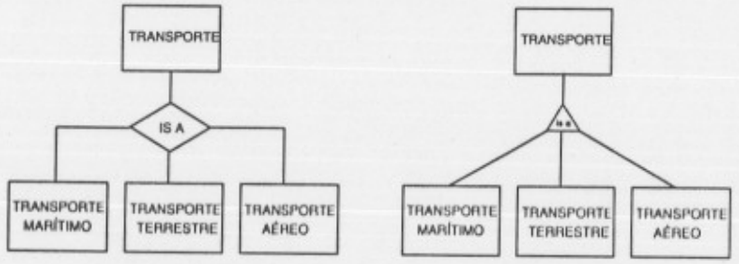


Notação Alternativa
(com semântica)

Estrutura de Especialização-Generalização

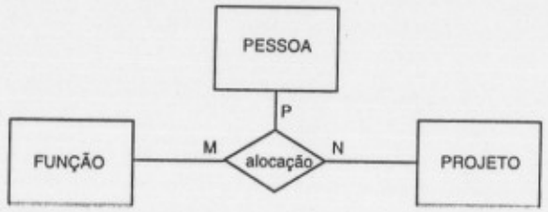
Em seu trabalho original, Chen não tratava esse tipo de elemento semântico. Entretanto, as evoluções introduzidas em seu método de representação acabaram por "se aproveitar" da notação utilizada para relacionamentos e implementar um tipo especial de estrutura: o relacionamento do tipo "IS A" (em inglês, "é um").

A estrutura "IS A" estabelece o vínculo de especialização entre a entidade generalizada e as entidades especializadas. Também pode ser utilizada a notação tradicional (um triângulo) para a representação de uma especialização. Essa é, inclusive, a notação atualmente utilizada pela ferramenta CASE da empresa de Chen.



Estrutura de Agregação

Não há tratamento específico para esse tipo de estrutura. Chen mostrava em seu trabalho original exemplos de relacionamentos ternários, tratando-os simplesmente como estruturas de relacionamento.



Qual a Melhor Notação?

Dentre as ferramentas CASE existentes no mercado podemos encontrar aquelas que são voltadas a uma única técnica de representação gráfica, outras que permitem um pequeno conjunto de técnicas e outras que permitem uma grande variedade de representações. Deve-se levar em consideração que aquelas que permitem o uso de várias técnicas de representação, algumas vezes, acabam por liberar o uso combinado de elementos de técnicas distintas em um mesmo diagrama.

Isso pode representar duas coisas: liberdade de criação de uma nova representação particular (o que não nos parece uma grande necessidade) e possibilidade de utilização indevida de símbolos nos diagramas.

Algumas vezes, a possibilidade de personalizar uma notação já existente poderá significar a diferença entre incluir a semântica necessária ou perdê-la. Por outro lado, deve-se ter em mente que:

> *Toda personalização ou alteração de critérios de representação já existentes nas técnicas conhecidas pode levar a erros de interpretação ou ao desconhecimento futuro do significado desses símbolos por outras pessoas.*

Caso isso ocorra, estaremos criando uma situação onde um novo dialeto passa a ser agregado a uma língua já disseminada e conhecida por vários povos. O que isso pode causar? O "Efeito Torre de Babel", ou seja, cada um falando sua própria língua e não sendo completamente compreendido por todos os outros.

Escolher entre uma ou outra notação dependerá de critérios e limitações impostos pelo próprio ambiente de desenvolvimento. Enfoque de projeto, ferramentas disponíveis, grau de conhecimento de uma ou outra técnica e vários outros pontos nos levarão a fazer a escolha por um ou outro método de representação. O ponto mais importante a ser destacado, todavia, é que:

> *Todos deverão procurar falar uma mesma língua e, mesmo quando isso não for possível, todos deverão ser capazes de se comunicar com os demais, seja com sua própria linguagem, com a linguagem do outro ou até através de uma terceira linguagem comum a ambos.*

Somente assim estaremos garantindo que a adoção de uma forma de representação gráfica sirva como elemento para comunicação de conceitos de modo mais ágil e sem ambigüidades o que, afinal, é o objetivo maior da modelagem conceitual.

Referências Bibliográficas

[1]CHEN, Peter P. The Entity-Relationship Model: Toward the unified view of data. mar./1976.

[2]MARTIN, James. *Técnicas Estruturais e CASE*. São Paulo: MacGraw Hill, 1991.

4

Derivação do Modelo Lógico

A obtenção de um modelo lógico a partir de um modelo conceitual pode ser feita pela aplicação de um conjunto de regras bem definidas. Essas regras atuarão, basicamente, em dois grupos distintos de elementos: as estruturas de relacionamento, agregação e especialização de um lado e as entidades e seus atributos de outro.

Veremos que, através de técnicas bastante objetivas, poderemos, partindo de um modelo conceitual, obter o modelo lógico desejado. As regras a serem apresentadas visarão à geração de um modelo lógico baseado na abordagem relacional. Também existem regras para a obtenção de modelos baseados na abordagem *rede-codasyl*. Entretanto, essas regras não serão alvo de nossa atenção em função de sua restrita aplicação no mercado e da constante intenção de não procurar apresentar conceitos desnecessários àqueles que se iniciam nas técnicas de modelagem.

Passos para a Derivação do Modelo Lógico

Muitos autores, em suas abordagens sobre o tema modelagem de dados, têm direcionado o processo de modelagem diretamente para o nível lógico. Partem, desde o início de seu processo, para a construção de um modelo fortemente dependente do ambiente onde será implementado. Isso tem levado à obtenção de modelos eficientes mas bastante dependentes da tecnologia que os orientou.

Dentro da proposta apresentada nos capítulos anteriores, vimos que a geração de um modelo lógico deveria ser antecedida pela obtenção de um modelo conceitual. Esse modelo conceitual deveria ser independente da tecnologia disponível, procurando aproximar-se o máximo possível do mapeamento fiel do ambiente observado.

A grande vantagem dessa proposta é que a partir do modelo conceitual gerado, poderemos aplicar regras predefinidas em função da tecnologia a ser empregada e, assim, obter os modelos necessários. Isso fará com que a partir de um mesmo modelo conceitual possamos gerar modelos lógicos para bancos de dados baseados na abordagem rede-codasyl, na abordagem "relacional" ou até mesmo em novas abordagens, tais como bancos de dados orientados a objetos.

Assim, podemos definir que o processo de obtenção de um modelo lógico a partir de um modelo conceitual segue os seguintes passos:

1. Obter o modelo conceitual.
2. Definir o tipo de implementação (rede-codasyl, relacional, O-O).
3. Aplicar as regras de derivação específicas.
4. Adaptar o modelo às necessidades.

Perceba que dentre os passos a serem seguidos existe, em último lugar, uma atividade extremamente importante para a geração de um modelo lógico aplicável efetivamente à solução de problemas práticos do dia-a-dia.

Adaptações no Modelo Lógico Derivado

A adaptação do modelo obtido pela aplicação dos passos 1 a 3 visa a criar estruturas de implementação que atendam às reais expectativas das aplicações. Hoje, cada vez mais, aspectos como desempenho, flexibilidade, independência e segurança têm sido impactantes na definição dos modelos lógicos a serem construídos. A inserção proposital de pequenas doses de redundância, a desnormalização, o seccionamento de modelos, a inserção de elementos adicionais para controle e facilidade de recuperação de dados têm sido, entre outros, os "artifícios" inseridos nos modelos adaptando-os para os aspectos práticos do dia-a-dia.

A imperfeição e as limitações da tecnologia a ser aplicada na implementação das estruturas de dados geradas nessa atividade fazem com que seja necessária, após a derivação, a execução desses ajustes adicionais. Um fato importante a ser considerado, entretanto, é que esses ajustes devem aparecer tão somente na fase final de "acabamento dos modelos". Considerá-los previamente, eventualmente até no modelo conceitual, seria extremamente indevido por inserir elementos não inerentes ao modelo desnecessariamente.

Outra justificativa para adaptações aos modelos gerados é a recente demanda por estruturas que atendam às necessidades de informações analíticas (agrupadas, seccionadas, particionadas, etc.) ou a sistemas de informação gerencial (SIG). Essas novas necessidades têm levado os projetistas de BDs à adoção de novas estratégias para a definição das estruturas lógicas dos dados. Conceitos como *data warehouse* (depósitos centralizadores de dados para aplicações não estruturadas) e *OLAP* (on-line analitical processing — processamento analítico de grandes volumes de dados feito em modo on-line) têm impactado de forma bastante significativa os paradigmas para a concepção das estruturas lógicas de armazenamento de dados. Não abordaremos essas estratégias neste livro por se tratar de elementos de maior complexidade que requerem, no mínimo, maior vivência na área de

modelagem. Também necessitaríamos explorar, nesse caso, conhecimentos específicos sobre recursos e técnicas recentemente disponibilizadas na área de banco de dados. Como não podemos assumir estarem esses conhecimentos disponíveis a cada um dos leitores, e como não haveria sentido em explorar aqui cada um deles, omitiremos esse tratamento.

Conceitos Utilizados na Derivação de Modelos Relacionais

Por ser a derivação do modelo lógico fortemente dependente dos conceitos e da tecnologia de implementação a ser utilizada, iremos abordar, de modo bastante objetivo, os principais conceitos envolvidos na abordagem relacional. Não esgotaremos o assunto, pois para isso seria necessário um trabalho bastante extenso. Como o objeto deste livro não é a abordagem relacional, e sim a modelagem de dados baseada no modelo relacional, indicaremos na bibliografia outros livros para sua consulta.

O modelo relacional, proposto já na década de 70 por Edgard F. Codd, foi concebido a partir da associação estabelecida entre os conceitos e regras que norteiam a teoria dos conjuntos e as possíveis estruturas de manipulação de dados até então utilizadas. Codd percebeu, naquele instante, que seria possível aplicar as operações conhecidas na matemática, para manuseio de conjuntos, sobre estruturas de dados. Essa abordagem garantiria, a partir daí, um método formal e rigoroso para o tratamento, até então desestruturado, das estruturas de dados. A criação, manipulação, combinação, seleção e tantas outras operações realizadas sobre arquivos de dados poderiam passar a ser demonstráveis e comprováveis pelas operações da teoria dos conjuntos.

A visão básica estabelecida era de que registros de um arquivo assemelhavam-se a tuplas de uma relação. Uma relação, por sua vez, assemelhava-se a um arquivo onde cada uma de suas tuplas, contendo elementos de domínios distintos, agregados de forma predefinida, podia ser entendida como um registro e seus campos.

Assim, uma relação R, definida sobre os domínios D1, D2 e D3, seria formada pelas tuplas representativas dos domínios D1, D2 e D3 e expressa através de um conjunto:

R = { d11,d21,d31; d12,d22,d32; d13,d23,d33; d14,d24,d34; ... }

Por definição, cada elemento da relação R deve ser único (não repetido na relação) e ter todos os domínios instanciados (não aceitando, por exemplo, a ausência de d21 e conseqüente geração de uma ocorrência "d11,d31").

Por exemplo, poderíamos ter quatro domínios, não necessariamente distintos, cada um deles com uma série de elementos. Ao definirmos a relação R como o conjunto formado pelo agrupamento desses domínios, poderíamos obter algo como:

Como pôde ser visto no exemplo acima, cada uma das tuplas da relação R tem instanciados os domínios D1, D2, D3 e D4, apesar de não necessariamente nessa ordem. Como a própria relação é definida como um conjunto de domínios, não há obrigatoriedade no estabelecimento de uma ordenação dos domínios mas sim em sua presença.

Entretanto, para fins de facilitação de visualização e entendimento, muitas vezes acaba-se por representar uma relação em modo tabular ou plano. Isso faz com que sua aparência se aproxime bastante daquela que originalmente temos em mente quando tratamos a questão de layouts de arquivos. Vamos representar agora a mesma relação vista anteriormente sob o modo tabular:

Relação R

Domínio D1	Domínio D2	Domínio D3	Domínio D4
Nome de Pessoa	Local de Nascimento	Cor de cabelo	Número de CPF
José	Manaus	Ruivo	004
Maria	Salvador	Preto	006
João	Manaus	Preto	001
José	São Paulo	Louro	003

Perceba que se analisarmos, agora sob o ponto de vista de estrutura de dados, temos uma seqüência de elementos (registros), cada um deles composto por uma seqüência de domínios (campos). Como foi possível estabelecer essa associação entre conceitos, também ficou provada a possibilidade de execução das operações de manuseio de conjuntos sobre as estruturas de dados.

Partindo-se dessa premissa, Codd elaborou o que se denomina hoje de *abordagem relacional* ou *modelo relacional*, trabalhando primeiramente

sobre os aspectos conceitais e formais do assunto e, depois, buscando a implementação através de software de seu modelo. Apesar de matematicamente perfeita, sua abordagem esbarrou, entretanto, em um problema significativamente sério: o desempenho das implementações de softwares relacionais. Isso se devia basicamente à necessidade do tratamento de grandes volumes simultâneos de dados. Como a abordagem da teoria dos conjuntos não está limitada a restrições de tempo, espaço e volumes, quando então podemos manusear, visualizar e investigar todo um conjunto independentemente de seu tamanho, passou a ser necessária também essa visibilidade para os softwares que implementavam a teoria relacional. É obvio que isso só poderia acabar, na década de 70, implicando sérias restrições de desempenho às aplicações que devessem manipular elevados volumes de dados simultaneamente.

Com o passar do tempo, entretanto, a disponibilização de novos recursos computacionais de larga escala fez com que a teoria relacional passasse também a ser viabilizada. Hoje, falamos em micros com 16 megabytes de memória para execução de aplicações mono-usuárias baseadas, por exemplo, em um software gerenciador de bancos de dados relacional do tipo MS-Access. Há 10 anos, muitas das grandes máquinas do tipo mainframe (processadores corporativos centralizados) não chegavam sequer a ter essa capacidade de memória para atender a redes com diversos usuários. Imagine então há 20 anos... Máquinas com menos de 1 megabyte de memória sendo cogitadas para um ambiente de processamento relacional realmente seriam um desafio ao bom senso.

Codd, por sua vez, não desistiu de sua abordagem. Ele sabia estar certo das idéias que defendia e por elas lutou, durante mais de 10 anos, contra os fornecedores de software, contra o mercado e contra o próprio governo americano, que sempre teve muita força no estabelecimento de padrões na área de informática. Esse é o depoimento do próprio Codd em seu novo livro em que apresenta o Relational Model 2 (RM/2), uma nova evolução sobre o modelo relacional. Felizmente, a evolução de hardware e software básico fez com que, hoje, o modelo relacional seja reconhecido como a "solução" para a abordagem de bancos de dados. Ainda existem deficiências mas, com certeza, cada dia estamos mais perto do estado-da-arte necessário para a implementação eficiente do modelo relacional. Máquinas com processamento paralelo maciço, processamento simétrico, sistemas operacionais multitarefa, discos de alto desempenho, memórias de alta capacidade, processadores mais velozes e tantas outras novidades nos aproximam do estágio tecnológico necessário.

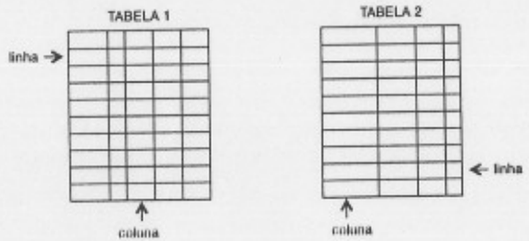
Implementação do Modelo Relacional

De acordo com a visão estabelecida por Codd, o modelo relacional passou a implementar as estruturas de dados, independentemente de seus

tipos, através de uma única visão: tabelas. O conceito de tabela era visão tabular ou plana representativa de uma relação. Daí, inclusive, advem o termo "relacional" que não tem nada a ver com os relacionamentos estabelecidos entre as diversas tabelas, mas sim com a implementação de *relações* através de uma estrutura de dados *relacional*.

Uma tabela, seguindo a definição de relação, seria composta de linhas ou tuplas, cada uma delas, por sua vez, sendo obtida a partir da instanciação de domínios predefinidos como formadores da relação. Esses domínios, para o modelo relacional, são tratados como colunas.

No modelo relacional, os dados são representados em forma de **tabelas** *(relações), ou seja, através de* **linhas** *(tuplas) e* **colunas** *(domínios).*



Conceitos

1. Chave Candidata e Chave Primária (PK — Primary Key). Somente utilizando a estrutura básica de linhas e colunas para a construção de tabelas não teremos ainda todos os elementos necessários para a implementação do modelo relacional. Como vimos, um dos conceitos básicos que definem uma relação é o fato de que "não poderão existir dois elementos iguais dentro de uma relação". Mas o que caracteriza um elemento de uma relação? Resposta: a instanciação de todos os seus domínios. Assim, no exemplo já utilizado anteriormente, podíamos ver o nome JOSÉ instanciando duas vezes o domínio NOME DE PESSOA, mas por estar acompanhado por valores diferentes nos demais domínios isso não caracterizava um elemento repetido na relação.

Domínio D1	Domínio D2	Domínio D3	Domínio D4
Nome de Pessoa	Local de Nascimento	Cor de Cabelo	Número de CPF
José	Manaus	Ruivo	004
Maria	Salvador	Preto	006
João	Manaus	Preto	001
José	São Paulo	Louro	003

Como essa condição de não-existência de elementos repetidos em uma tabela é uma necessidade básica a ser cumprida para o estabelecimento de um modelo relacional realmente aderente aos conceitos de "relação", foi introduzido o conceito de *CHAVE IDENTIFICADORA da tabela*. Sabemos que se o conjunto total dos valores instanciadores dos diversos domínios que compõem a tabela não se repetem, então não temos linhas iguais ou duplicatas. Entretanto, analisar todos os valores de cada um dos domínios pode ser, em alguns casos, bastante complexo e demorado.

Codd estabeleceu uma simplificação que visa a facilitar essa análise. Se, por análise prévia e conhecimento do ambiente que está sendo modelado, formos capazes de identificar domínios nos quais sabemos que a existência de valores instanciados não será repetida, podemos eleger esses domínios, ou colunas, como CANDIDATAS A CHAVE. A simples análise e comprovação de diferença dessa coluna, comparando seu valor com os demais valores existentes nas demais linhas, fará com que seja possível garantir a não duplicidade de toda uma linha.

Veja em nosso exemplo o caso da coluna NÚMERO DO CPF. Sabemos, por definição, que "nome de pessoa", "local de nascimento" e "cor de cabelo" são colunas que podem "potencialmente" ter seus valores repetidos em diferentes linhas. Pessoas com mesmo nome são comuns, nascimentos no mesmo lugar são esperados e cores de cabelo repetidas um fato normal. Já se analisarmos a coluna NÚMERO DO CPF veremos que, por definição (para a Receita Federal, ao menos), não devem existir duas pessoas com o mesmo CPF. Em outras palavras, se analisarmos nossa tabela e percebermos que duas linhas contêm CPFs diferentes, então serão obrigatoriamente linhas inteiramente diferentes.

Perceba que a chave, ao mesmo tempo em que facilita a análise e diferenciação, também impõe restrições. Se desejarmos que duas pessoas compartilhem o mesmo CPF (marido e mulher, por exemplo), deveremos mudar nossa chave para que, além de CPF, conste também o NOME DA PESSOA como formadora da chave. Assim, a dupla de colunas deverá sempre ser analisada e, mesmo com CPFs repetidos, não estaremos caracterizando mais uma duplicidade, desde que os NOMES DE PESSOA sejam diferentes.

Durante a análise de uma tabela poderemos encontrar diversas colunas CANDIDATAS A CHAVE. Basta que exista mais de um domínio que apresente a característica de não repetir seus valores durante as instanciações para todas as linhas da tabela. Dentre essas várias colunas deveremos escolher uma e elegê-la como CHAVE IDENTIFICADORA ou CHAVE PRIMÁRIA (PK — *Primary Key*) da tabela em questão. Já que qualquer uma das colunas CANDIDATAS pode ser escolhida, não haveria sentido manter mais de uma como chave. Isso somente aumentaria a complexidade de análise durante o processo de identificação de possíveis duplicatas. Assim, basta que seja escolhida uma das CHAVES CANDIDATAS e atribuída a ela a função de CHAVE IDENTIFICADORA. Vamos analisar um exemplo de uma tabela de ELEITORES concebida junto ao ambiente do Tribunal Regional Eleitoral. Nessa tabela, poderíamos ter a seguinte situação:

- COLUNAS:

  Nome do Eleitor, Data de Cadastramento, Junta Eleitoral, Seção Eleitoral, Número do RG, Número do CPF, Número do Título de Eleitor, Data de Nascimento, etc.

- CHAVES CANDIDATAS:

  Número do Título de Eleitor, Número do RG e Número do CPF

- **CHAVE IDENTIFICADORA ou CHAVE PRIMÁRIA (PK — Primary Key):**

  Número do Título de Eleitor

Nesse exemplo, assumimos que o Número do Título de Eleitor seria a chave candidata escolhida, pois é natural ao ambiente no qual será utilizada. Isso implica que devemos garantir a unicidade de valor desse número em toda a tabela de ELEITORES a ser criada. Isso poderá ser feito através de recursos do próprio gerenciador de bancos de dados ou, até mesmo, através de consistência da própria aplicação. É claro que deixar essa restrição a cargo da própria aplicação não é uma solução recomendada. Como a própria implementação relacional obriga a unicidade de chave primária, os próprios SGBDRs (Sistemas Gerenciadores de Bancos de Dados Relacionais) implementam essa restrição de modo automático durante a definição da estrutura de uma tabela.

O conceito de chave primária não deve ser estranho para nenhuma daquelas pessoas que já tenham tido convivência com sistemas baseados em técnicas de recuperação convencional (arquivos indexados, por exemplo). Isso facilita o entendimento do conceito de chaves assumindo valores únicos mas, por outro lado, acaba induzindo, às vezes, ao estabelecimento de uma analogia não verdadeira.

A chave primária em um arquivo indexado, por exemplo, é um recurso que, além de garantir unicidade de valores entre dois ou mais registros, possibilita o acesso direto a uma informação desejada. Apesar de sabermos que esse acesso na verdade não é direto, mas sim feito através de um mecanismo de índices e uma posterior busca seqüencial, traduz-se a funcionalidade de acesso direto durante sua utilização. Assim, o uso da chave primária é estabelecido como o modo para recuperação de dados específicos dentro de um arquivo convencional.

Já em uma tabela, que é a estrutura de implementação de uma relação, não se pode pretender estabelecer vínculos com métodos de acesso. Se a tabela é a implementação de uma relação, e a relação, por sua vez, é um conceito genérico associado a uma coleção de tuplas, formadas por seqüências de valores de domínios, não há por que estabelecer métodos de acesso para uma relação. As tuplas de uma relação simplesmente existem e são manipuláveis. São localizáveis por qualquer um dos valores existentes em cada um de seus domínios. Assim, se tivermos algo parecido com a tabela a seguir, poderemos recuperar qualquer uma das tuplas por qualquer um dos domínios existentes:

Placa	Cor	Potência	Modelo	Ano de Fabricação
AAA-1010	Cinza	96 HP	SL	1995
ACB-3431	Preto	80 HP	CL	1994
ABC-7438	Azul	75 HP	SL	1996

Todas as buscas listadas a seguir seriam válidas:

"Recupere o carro produzido em 1995" traria a tupla número 01.
"Recupere o carro de cor azul" traria a tupla número 03.
"Recupere o carro de placa ACB-3431" traria a tupla de número 02.
"Recupere o carro de modelo SL" traria a tupla de número 01 e 03.

A localização dos registros seria feita independentemente de a coluna ser definida como chave primária ou não. Esse é o conceito relacional que tanta flexibilidade traz para o processo de tratamento de estruturas de dados.

Com isso se conclui que a finalidade da chave primária, no modelo relacional, é desvinculada do fato de podermos vir a recuperar linhas, ou ocorrências de uma tabela, através de sua utilização. Como já vimos a chave primária serve, basicamente, para estabelecer a unicidade de uma tupla dentro da relação. No tópico a seguir, veremos que essa chave servirá, também, como meio para o estabelecimento de relacionamentos entre tuplas de diferentes relações.

2. Chave Estrangeira (FK — Foreign Key). Como vimos na fase de modelagem conceitual, temos, basicamente, três elementos para representar um ambiente observado: as entidades, os atributos e os relacionamentos. Através de tabelas, portanto, deveremos ser capazes de implementar todos esses elementos. Com os conceitos já vistos até o momento (tabelas, linhas e colunas) poderíamos implementar, facilmente, os dois primeiros elementos: as entidades e seus atributos.

Para que possamos, entretanto, implementar relacionamentos entre entidades deveremos passar a tratar mais um conceito importante no ambiente relacional: as chaves estrangeiras.

Uma chave estrangeira recebe esse nome para denotar algumas características importantes:

- Ela é uma chave e, portanto, identifica de modo único uma tupla.
- Ela não está em seu local original (local de origem), mas sim em um local para onde foi migrada (no estrangeiro).
- Ela tem um local de origem e lá possui suas características originais (é uma chave primária).

Perceba que, pelas características anteriormente descritas, uma chave estrangeira nada mais é do que uma chave primária de uma tabela que aparece repetida em outra.

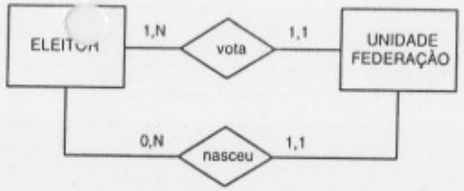
Veremos mais adiante que, resumidamente, esse é o método para o estabelecimento de relacionamento entre duas tabelas. Basta que se pegue a chave primária de uma das tabelas e a agregue em outra. Essa chave, quando criada em outra tabela, passará a ser uma *chave estrangeira*.

Se para alguns o termo "chave estrangeira" soava estranho até hoje, talvez agora passe a fazer sentido. O processo de migração de chaves é o responsável por esse termo, tornando-o justificável.

Exemplo:

Imaginemos, em continuidade a um exemplo já fornecido, que tenhamos modelado o ambiente de um processo eleitoral. Identificamos entidades tais como: ELEITOR, UNIDADE DA FEDERAÇÃO (Estados e Territórios), entre outros. Sabendo que as entidades têm seus atributos próprios, definimos como chave primária da tabela ELEITOR a coluna NÚMERO DO TÍTULO DE ELEITOR e como chave primária da tabela UNIDADE DA FEDERAÇÃO a coluna SIGLA DA UF.

Identificamos pelo modelo que existem dois relacionamentos entre as entidades. O primeiro diz respeito à UNIDADE DA FEDERAÇÃO onde o ELEITOR vota. O segundo diz respeito à UNIDADE DA FEDERAÇÃO onde o ELEITOR nasceu.



Nesse caso, obteríamos as seguintes tabelas a implementar (já com seus atributos, chaves e relacionamentos):

- **Tabela UNIDADE DA FEDERAÇÃO**

Chave Primária (PK)	: SIGLA-UF
Colunas	: Nome, Area-m2

- **Tabela ELEITOR**

Chave Primária (PK)	: NUM-TÍTULO-ELEITOR
Colunas	: CPF, RG, Nome, Estado-civil
Chave Estrangeira (FK)	: SIGLA-UF-VOTO
Chave Estrangeira (FK)	: SIGLA-UF-NASCIMENTO

Assim, podemos, pelas chaves estrangeiras SIGLA-UF-VOTO e SIGLA-UF-NASCIMENTO, relacionar a tabela ELEITOR com a tabela UNIDADE DA FEDERAÇÃO. Se quisermos identificar o nome da unidade da federação (Estado ou Território) onde alguém vota, poderemos acessar a tabela UNIDADE DA FEDERAÇÃO a partir do conteúdo existente na coluna SIGLA-UF-VOTO.

3. Domínios de Atributos. O conceito de domínio de um atributo, ou de uma coluna, é fortemente associado ao conceito de domínios de relações. A idéia principal é que existe um conjunto de valores distintos que podem ser atribuídos a uma coluna (um a cada tupla). Entretanto, sabemos que pela diferente natureza de cada uma das colunas poderemos ter diferentes valores sendo válidos às instanciações.

Observe no caso da tabela de UNIDADES DA FEDERAÇÃO como cada uma das colunas pode receber valores bastante diferentes. A coluna SIGLA-UF, que por acaso é a chave primária da tabela, somente pode assumir valores preestabelecidos. Cada unidade da federação, quando é criada, recebe um nome, tem sua área delimitada e passa a ter uma SIGLA ou abreviatura para ser utilizada, por exemplo, no endereçamento postal.

Para a coluna SIGLA-UF, portanto, poderíamos atribuir uma das seguinte siglas: RS, SC, PR, SP, RJ, MG, BA, ... até MA, RO, RR. Isso é o que chamamos de um DOMÍNIO DISCRETO. Sabemos previamente qual é o conjunto permitido de valores inclusive em suas quantidades.

Já para a coluna AREA-M2, poderíamos ter valores variando entre 1 e N metros quadrados. Poderíamos determinar o valor de N através de uma estimativa do tipo: uma unidade da federação nunca terá um território maior que a maior região do país. Independentemente de não estabelecermos agora qual seria o exato valor de N, assuma que existe um "valor inicial" e um "valor final" válido para a atribuição à coluna AREA-M2. Perceba, entretanto, que, apesar de sabermos os limites mínimo e máximo, não podemos, ou não queremos, elencar cada um dos possíveis valores existentes. Eles caracterizam uma faixa contínua de valores. Seria desnecessário elencar cada um deles. Isso caracteriza um DOMÍNIO CONTÍNUO.

Aparentemente, seriam somente essas duas modalidades de domínios as existentes. Se um domínio não é contínuo, logo será discreto e vice-versa. Entretanto, existe, para fins de modelagem lógica relacional, um outro tipo de domínio que nos interessa. Ele é responsável por grande parte das discussões e problemas gerados quando o assunto discutido se refere a "domínios".

Estamos falando do DOMÍNIO NULO. Se estabelecermos uma correlação com a teoria dos conjuntos, um domínio nulo seria representativo de algo como um conjunto vazio. Sabemos que se há um conjunto é porque, teoricamente, existem elementos envolvidos. Por outro lado, assumimos como verdadeira a condição de que um conjunto pode ser vazio.

Assim, alguns atributos, ou colunas, terão seu domínio definido como NULO. O valor aplicável para a instanciação desse atributo é "nulo". Perceba que não estamos falando em um valor 0 (zero) ou " " (branco). Esse é o grande ponto de importância quando se procura caracterizar um valor nulo:

> *Um valor NULO não é igual a 0 (zero) ou " " (branco ou espaço). Ele, na verdade, não é igual a nada, pois simplesmente não existe! Não podemos comparar um elemento inexistente a algo que efetivamente exista tal como um caracter " " (branco) ou o número 0 (zero).*

Isso leva a uma série de implicações práticas do ponto de vista de tratamento em nível de linguagem de programação, gerenciadores de bancos de dados, comparações de strings, comparações de valores, etc. Para muitas situações de projeto de estruturas lógicas será extremamente importante reconhecer a existência e o significado de um valor NULO em uma certa coluna de uma tabela.

Atualmente, a teoria relacional (proposta por Codd) tem tratado indistintamente dois tipos de NULIFICAÇÃO (existência de um valor NULO). Na nova proposta de Codd para a teoria relacional (o RM/2 — Relational Model 2), já passarão a ser distintos esses dois tipos de NULOS.

Apesar de ainda não estarmos sob as normas do RM/2 será importante, para nosso processo de derivação de modelos lógicos, reconhecer que podemos ter dois tipos de valores NULOS:

1. Nulos Inaplicáveis. São os valores NULOS existentes em uma coluna em função da não aplicabilidade dessa coluna para a linha em questão. Isso significa que temos uma coluna, representativa de um atributo de um objeto, que em algumas linhas (tuplas) não tem significância. Não pode ser valorado, pois simplesmente não se aplica a esse elemento que está sendo instanciado.

Exemplo:

Suponha que tenhamos modelado um ambiente onde percebemos a existência de EMPREGADOS de EMPRESAS. As EMPRESAS têm seus atributos, assim como os EMPREGADOS. Dentre eles poderíamos citar:

EMPRESA:

CGC, Inscrição Estadual, Razão Social, Nome Fantasia, Endereço, Tipo

EMPREGADO:

RG, Nome, Sexo, Data Nascimento, Qtde. de Gestações, Endereço

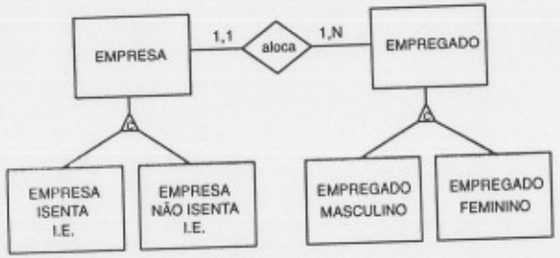
Perceba que alguns dos domínios são discretos (Tipo, Sexo), outros são contínuos (RG, CGC, Data de Nascimento) e outros são NULOS INAPLICÁVEIS para algumas instâncias (Inscrição Estadual e Quantidade de Gestações).

No caso específico das EMPRESAS isentas de Inscrição Estadual, não haverá um valor a ser alocado à coluna referente a esse atributo quando se estiver armazenando essas ocorrências. Também no caso de EMPREGADOS que sejam do sexo masculino, não haverá sentido manter a coluna QTDE. DE GESTAÇÕES com qualquer que seja o valor.

Nesses dois casos, apesar de serem numéricos os domínios das colunas INSCRIÇÃO ESTADUAL e QTDE. DE GESTAÇÕES, não seria correto deixá-las com o valor 0 (zero). Isso representaria que o número de INSCRIÇÃO ESTADUAL de algumas empresas seria 0 (zero), o que além de errado faria com que mais de uma empresa tivesse o mesmo número. Também no caso dos EMPREGADOS do sexo masculino, deixar a coluna QTDE. DE GESTAÇÕES com o valor 0 (zero) representaria que "ele" não engravidou nenhuma vez. Será que um dia ainda engravidará? Apesar dos avanços da medicina, parece ser realmente um dado inaplicável. Perceba que o mesmo valor 0 (zero) na mesma coluna, quando o EMPREGADO for do sexo feminino, passará a ser válido e representará efetivamente que

uma empregada não desenvolveu, ainda, uma gestação até o momento. Apesar de algumas, por opção ou acaso, não virem nunca a desenvolvê-la, a indicação 0 (nenhuma gestação) passa a ser aplicável e correta a essas instâncias.

Do ponto de vista de modelo de dados conceitual, essa característica pode denotar a existência de entidades sem a devida especialização (estrutura de Generalização-Especialização). Se temos EMPRESAS ISENTAS e EMPRESAS NÃO ISENTAS de INSCRIÇÃO ESTADUAL, talvez isso seja um sinal de que uma estrutura de especialização fosse recomendada para demonstrar essa situação. Se temos EMPREGADOS do sexo masculino e do sexo feminino e queremos para eles diferenciar atributos aplicáveis ou não, poderíamos também utilizar uma estrutura de Generalização-Especialização.



Do ponto de vista lógico, pelo que determinava a primeira forma normal anteriormente (veremos isso mais a frente), não deveriam existir valores NULOS em uma tupla de uma RELAÇÃO. Isso era bastante taxativo e criava situações complexas para a implementação das tabelas. Hoje a realidade é outra. A convivência com valores NULOS em colunas de uma tabela já é possível e até bastante natural. A própria sintaxe de definição das tabelas (SQL) já prevê uma cláusula para a definição de colunas que aceitam ou não aceitam valores NULOS. Sabe-se que, do ponto de vista prático, não haveria sentido implementar sempre uma tabela para EMPREGADOS MASCULINOS e outra para EMPREGADOS FEMININOS só por causa da existência de um ou outro atributo exclusivamente aplicável a uma ou outra situação. Assim, o que acabaremos por ter finalmente será uma única tabela EMPREGADOS, onde a coluna QTDE. DE GESTAÇÕES poderá assumir o valor 0, valores diferentes de zero ou o valor NULO.

2. Nulos Desconhecidos. Outra situação onde poderemos encontrar valores NULOS em colunas de tabelas é quando um certo valor "ainda" não for conhecido. Perceba que frisamos bem o termo "ainda". Significa que essa coluna não teve um valor estabelecido até o momento, mas que ele existe e é aplicável.

Essa situação pode ser caracterizada quando, por restrições de disponibilidade de informações, não pudermos estabelecer o valor devido para uma dada coluna durante seu processo de atualização. Vamos utilizar o próprio exemplo anteriormente referenciado. Dentre os atributos identificados para a entidade EMPREGADO temos o ENDEREÇO. Suponha, entretanto, que um EMPREGADO que acabou de ser contratado pela EMPRESA esteja em processo de recrutamento e não saiba informar, no instante de seu cadastramento, o endereço no qual estará habitando. Pode ser um fato atípico alguém não saber o nome da rua onde mora, mas para alguém que recentemente mudou-se para uma nova cidade isso pode ser até compreensível.

Deixando de lado os motivos que podem levar alguém a não saber o endereço onde mora, voltemos ao que nos interessa. O fato relevante é que, quando do cadastramento do EMPREGADO, não teremos a informação do endereço para fornecer como valor a ser alocado à coluna ENDEREÇO. Teremos duas alternativas: ou não cadastramos o EMPREGADO até que ele tenha todos os dados para nos fornecer ou cadastramos somente os dados de que dispomos naquele momento. Se optarmos por cadastrar um EMPREGADO sem a informação de seu ENDEREÇO, teremos nessa coluna o valor NULO. Esse é um tipo de NULO que caracteriza NULO DESCONHECIDO. Veja que o dado ENDEREÇO é aplicável a todos os EMPREGADOS, mas que para alguns deles ele pode ser DESCONHECIDO.

Alguns gerenciadores de bancos de dados já têm disponíveis recursos para que possamos caracterizar na base de dados uma coluna que contenha um valor NULO DESCONHECIDO. Pode-se, em alguns casos, definir um string a ser alocado como default sempre que não se tenha o dado correto a ser alocado a uma coluna. Esse string pode ser uma seqüência de caracteres predefinidos, uma palavra, espaços em branco, o valor 0, etc. Deve-se escolher, caso se deseje utilizar esse recurso, um string que não seja possível de ser fornecido como dado pelo próprio processo de obtenção de informações para cadastramento. Isso poderia levar a confusão pois, após certo tempo, não saberíamos se o string foi fornecido pelo processo de obtenção de informações ou gerado automaticamente pelo gerenciador de bancos de dados para indicar um valor desconhecido.

Hoje, pelas regras de definição de colunas adotadas pelo padrão SQL, todas as colunas, com exceção daquela, ou daquelas, que formam a chave principal de uma tabela, podem assumir o valor NULO. Quando desejarmos que uma coluna não possa assumir o valor NULO, devemos instruir isso ao gerenciador de bancos de dados.

Regras de Derivação

Como citamos anteriormente, a obtenção de um modelo lógico deveria ser feita a partir de um modelo conceitual previamente gerado. Para

tanto, deveríamos dispor de uma série de regras de derivação : fossem aplicadas sobre o modelo conceitual e que o transformassem, em função da topologia de BD usada, em um modelo relacional, rede-codasyl ou até hierárquico.

Veremos, a partir deste instante, portanto, essas regras. Como nosso objetivo, neste livro, é explorar a derivação de modelos relacionais, teremos, durante o processo de derivação, a utilização de uma série de conceitos e determinações impostas pela teoria relacional. Os conceitos necessários, tais como tabelas, linhas, colunas e chaves identificadoras, já foram apresentados neste capítulo. Outros conceitos adicionais que sejam necessários serão apresentados devidamente no instante de suas aplicações.

O processo de obtenção de um modelo relacional é baseado na execução de três atividades distintas: normalização das estruturas de dados, derivação das agregações e das generalizações-especializações e derivação dos relacionamentos.

1. **Normalização das Estruturas de Dados.** Esta atividade, imposta pelas regras de construção de estruturas relacionais, muitas vezes tem sido vista como elemento complicador do processo de modelagem lógica. Ela depende de critérios pessoais de interpretação e avaliação das estruturas de dados, e mesmo que alguns produtos digam poder fazê-la, automaticamente, não pode ser, de fato, realizada de modo automático. Os produtos que dizem fazê-la automaticamente somente automatizam algumas das atividades braçais executadas durante a normalização. A atividade mais importante, e crítica, que é a análise e definição de pertinências dos atributos a cada uma das tabelas, continuará a depender integralmente da intervenção manual do modelador. Vendo mais a frente o processo de normalização você entenderá melhor o que estamos afirmando.

2. **A Derivação de Estruturas de Agregação e Generalização-Especialização** pode ser feita, sob certos aspectos limitantes, de modo automático. Se desejarmos, entretanto, ter maior liberdade sobre a definição de estratégias de projeto, deveremos executar, também, essa atividade manualmente. Assim, poderemos analisar questões subjetivas, que muitas vezes poderiam passar despercebidas em um processo automático, e definir o tipo de derivação caso a caso, sempre buscando a geração de estruturas mais adequadas à futura implementação.

3. **A Derivação de Relacionamentos** é bastante simples e, caso você tenha a possibilidade de aplicação de alguma ferramenta CASE, poderá ser feita automaticamente. Apesar disso, abordaremos sua execução do ponto de vista manual pois, ao menos, deveremos ter consciência do que, como e por que, algumas derivações são feitas pelas ferramentas CASE.

Normalização

O processo de normalização tem sido visto por muitas pessoas apenas como um formalismo necessário à implementação de estruturas relacionais. Esse formalismo, quando não compreendido em sua finalidade, acaba por transformar-se mais em um fator complicante do que em um fator otimizador durante o projeto de bancos de dados.

Definitivamente a normalização não é algo simples de ser feito se não tivermos controle sobre as atividades que estejam sendo executadas, se não soubermos o que, como e por que estamos agindo. Quando esses pontos passam a estar claros, aí então passamos a efetivamente encarar o processo de normalização como uma alavanca para o processo de construção de bancos de dados.

O primeiro ponto importante a ser definido para que se tenha a correta visão desse processo é de que:

A normalização não é um processo com finalidade restritiva mas sim com caráter organizativo.

Encarar a normalização como imposição de regras que simplesmente restringem nossa liberdade no projeto de bancos de dados é ter uma visão muito simplificada do que representa esse processo. Acima de tudo, a normalização propõe uma série de passos a serem seguidos para que tenhamos um processo de verificação, validação e ajuste das estruturas de dados que porventura tenham sido observadas, modeladas mas que ainda apresentem distorções quanto à realidade.

Durante a modelagem conceitual poderemos estar trabalhando sobre estruturas não normalizadas. Essa é uma característica que muitas vezes tem passado despercebida por algumas pessoas: não há qualquer restrição sobre o fato de que uma entidade conceitual não se apresente normalizada. Lembre-se de que falamos em atributos referenciais, citando que estes indicavam pertencer a outras entidades que não aquela em questão. O fato de existir um atributo referencial indica claramente que estamos trabalhando em um ambiente não normalizado.

Também o fato de muitas vezes termos necessidade de representar um ambiente de escopo bastante grande (um modelo corporativo, por exemplo) nos levará obrigatoriamente a trabalhar com entidades não normalizadas. Normalizar um modelo corporativo acabaria por transformá-lo em um modelo de proporções incabíveis fazendo, então, com que ele acabasse por perder sua finalidade que é a de espelhar de modo geral um grande escopo através de um reduzido número de elementos. Lembre-se de que não é possível representar o mapa do Brasil com todas as suas ruas, quadras e lotes.

Outro fato que tem nos chamado a atenção é o fato de que muitas pessoas acreditam que a normalização é um processo que é realizado somente após definidas as prováveis tabelas de um sistema. Acreditam que a normalização ocorre somente como um elemento posterior ao projeto inicial de um banco de dados. Isso, apesar de verdadeiro, não é toda a verdade. A normalização pode ocorrer em três instantes distintos:

1. Durante o processo de concepção do modelo conceitual.
2. Durante a derivação do modelo lógico.
3. Após a derivação do modelo lógico.

Vejamos cada um deles, em detalhes:

Normalização durante o Processo de Concepção do Modelo Conceitual

Já dissemos claramente neste mesmo capítulo que o modelo conceitual não precisa obrigatoriamente estar normalizado. Isso entretanto não quer dizer que ele não possa estar normalizado em alguns casos. O grau de normalização que ele pode apresentar dependerá de vários fatores como, por exemplo:

- **Abrangência do modelo.** Quanto mais abrangente (em termos de escopo) for um modelo, mais chances, caso ele tenha sido adequadamente concebido, haverá de que ele tenha um baixo grau de normalização. Por outro lado, quanto menor for o escopo, quanto mais restrito for o ambiente a ser modelado, mais chances haverá de que tenhamos um modelo mais normalizado. Assim, vemos que, variando entre esses dois extremos, teremos diversos graus de normalização possíveis existindo em um modelo conceitual. Isso fará com que, em muitos casos, durante o processo de derivação do modelo conceitual já estejamos partindo de uma realidade normalizada, tendo então "antecipado" esse processo na fase conceitual.

- **Facilidade de abstração e compreensão.** O correto entendimento dos elementos presentes em um ambiente a ser modelado poderá nos levar mais facilmente à concepção de um modelo normalizado. Quando más interpretações ou abstrações indevidas estão presentes durante a fase de modelagem conceitual, acaba-se por produzir modelos que não atenderão à segunda e terceira formas normais. Felizmente as regras definidas para a segunda e terceira formas normais nos levarão a perceber as anomalias existentes por identificações indevidas de atributos e entidades. Através da aplicação dessas regras poderemos sanar as deficiências existentes no próprio modelo conceitual, separando-se entidades distintas que estejam mapeadas como única entidade. Lembre-se de que, como vimos no item anterior, muitas vezes essa característica não será uma deficiência de modelagem mas sim um elemento criado intencionalmente.

- **Conhecimento prévio do ambiente.** O fato de que já temos conhecimento prévio do ambiente a ser modelado, e assim do próprio resultado a ser obtido, tanto no nível conceitual como no nível lógico, nos induzirá, por mais que não queiramos, a previamente "enxergar" o ambiente de modo normalizado. Isso poderá em alguns casos ser benéfico e em outros nem tanto. Caso o escopo seja restrito teremos, de certo modo, o benefício de já partirmos para um modelo conceitual próximo do modelo normalizado. Se o escopo for abrangente, correremos o risco de tentar introduzir elementos em demasia no modelo por estarmos já visualizando esse "grande ambiente" de modo normalizado. Em ambos os casos, entretanto, deveremos tomar cuidado com o fato de que a semântica conceitual não deverá ser sacrificada em troca de excessivos elementos lógicos (normalizados) introduzidos durante a fase de modelagem conceitual. Essa é uma tendência de todos aqueles que têm uma visão muito "orientada a banco de dados" durante o processo de modelagem conceitual. Portanto, tome cuidado: Utilize seu conhecimento prévio mas não seja conduzido por ele.

Normalização durante a Derivação do Modelo Lógico

Caso você tenha concebido seu modelo conceitual e agora pretenda transformá-lo em um modelo lógico para implementação em ambiente relacional, você deverá ter certeza de que ele respeita os preceitos da tecnologia relacional. A existência de estruturas de dados normalizadas é um desses preceitos e não poderá ser violado, sob risco de colocar em cheque toda a futura implementação.

A própria derivação de um modelo conceitual para modelo lógico enfoca a questão da normalização. Obrigatoriamente deveremos atuar, durante a derivação, sobre as estruturas das entidades e dos relacionamentos, ajustando-os para a implementação relacional. Caso você, intencionalmente ou não, ainda não tenha feito qualquer tipo de normalização sobre as estruturas de dados das entidades, esta será a hora. Veremos, mais adiante, as regras a serem aplicadas nesse processo.

Pode-se dizer que deixar a preocupação com a execução do processo de normalização para esse instante é o ideal. Se nos preocuparmos antecipadamente estaremos, possivelmente, introduzindo dificuldades adicionais no processo de modelagem conceitual; se deixarmos para depois, teremos um processo de ajuste das estruturas das tabelas já concebidas que também levará a preocupações adicionais.

Normalização após o Processo de Derivação do Modelo Lógico

Essa terceira opção, na verdade, não se caracteriza como uma opção sistemática. Ela é, muito mais, um processo corretivo. Isso significa que, caso você não execute o processo de normalização em um dos outros dois instantes já abordados, acabará por criar tabelas cuja estrutura não está normalizada, o que resultará em deficiências e anomalias de projeto que, mais cedo ou mais tarde, terão de sofrer os ajustes necessários.

Veremos durante a abordagem das três formas normais que o fato de uma tabela não se encontrar normalizada traduz, sob o ponto de vista de sistema de informações, anomalias que comprometem processos, integridade, disponibilidade de informação, etc. Assim, normalizar uma tabela não é só uma possibilidade mas sim uma necessidade se desejarmos ter sistemas, ao menos, "bem comportados". Acreditar que se pode conviver com tabelas não normalizadas de modo sistemático é uma idéia equivocada. Seus sistemas de informação terão pouca ou nenhuma chance de sobreviver sem estruturas de dados normalizadas. Há de se ressaltar um ponto: Quando falamos em "desnormalizar tabelas" para fins de desempenho, ajuste ou outra finalidade qualquer, não estamos falando em não normalizar. Estamos sim falando em após termos normalizado uma tabela, regredi-la, intencionalmente, a uma forma não normalizada. Perceba que desnormalizar implica antes normalizar. Existem casos onde isso é possível, dentro de limites e controles seguros. Não é uma operação simples e requer conhecimento para que possa ser executada; portanto, tenha critérios e use o bom senso para avaliar em que situações se justifica esse procedimento.

Benefícios da Normalização

Dentre os vários pontos que poderiam ser citados como benefícios diretos do processo de normalização podemos citar alguns que têm exemplos práticos a serem apresentados e que, com isso, podem demonstrar sua efetividade.

1. **Estabilidade do Modelo Lógico.** A implementação de tabelas não normalizadas faz com que a estabilidade do próprio modelo lógico representativo dessas tabelas venha a ser comprometida. Por estabilidade entendemos a capacidade de um modelo manter-se inalterado face a mudanças que venham a ser percebidas ou introduzidas no ambiente que tenha sido modelado.

Caso as mudanças percebidas ou ocorridas no ambiente modelado venham a se refletir sobre o modelo lógico e, por conseguinte, sobre as tabelas definidas para sua implementação, teremos reflexos em diversos níveis. Do ponto de vista de sistemas de informação, poderemos ter impactos sendo causados sobre caminhos de navegação e acesso a tabelas, origem e destino de itens de dados, métodos de obtenção de informações, etc. Já

do ponto de vista de banco de dados, teremos impactos sobre as estruturas de armazenamento, alocação de espaços, previsão de taxas de crescimento, etc. Todas essas mudanças acabarão, em menor ou maior grau, por afetar o desempenho e a produtividade do processo de desenvolvimento de aplicações. Sabemos que hoje buscam-se meios para isolar as aplicações de mudanças ocorridas no meio externo, tentando com isso obter o maior grau de estabilidade possível durante o ciclo de desenvolvimento de sistema e, assim, evitando os efeitos indesejáveis de um processo cíclico do tipo *waterfall*.

Imagine um caso bastante simples onde um modelo (e suas tabelas) tenha sido concebido desnormalizado já em sua primeira forma normal. Mesmo que você ainda não tenha conhecimento exato da "Primeira forma normal" assuma que ela, em seu mais simples e restrito formato estabeleça que "uma tabela não deve ter itens de repetição". Veremos mais adiante que existem outros pontos envolvidos mas, por hora, nos basta para exemplificar essa definição resumida.

Vamos analisar uma única tabela de um modelo, em que não tenha havido a devida normalização:

Tabela FUNCIONÁRIO

Colunas:
Nome
Data de Nascimento
Endereço
Valor de Contribuição Sindical (repetindo-se 5 vezes)
Valor do Salário Básico
Telefone
Ramal

O fato de mantermos um "item de repetição" (o valor de contribuição sindical) na estrutura da tabela faz com que essa tabela não obedeça a uma das restrições impostas pela primeira forma normal. Isso por si só, se analisado isoladamente, poderia não nos dizer muita coisa de prático. Entretanto, vejamos como a estabilidade lógica de dados e processos pode ser afetada:

- **Impactos sobre a estabilidade lógica das aplicações.** O fato de possuirmos cinco repetições de um mesmo campo teria de ser mapeado logicamente por uma aplicação utilizando algum recurso disponível na linguagem de manipulação. Esse recurso normalmente é um vetor. O vetor é um elemento no qual coexistem diversos elementos, cada um alocando um espaço dentro de uma estrutura de dados logicamente contígua, todos tendo um mesmo

nome e cada um sendo referenciado por sua posição relativa dentro dessa estrutura.

vetor do campo VALOR DE CONTRIBUIÇÃO SINDICAL

valor (1)	valor (2)	valor (3)	valor (4)	valor (5)

Imagine que um vetor, previamente definido para ter cinco ocorrências, e que esteja sendo manipulado por uma série de aplicações distintas, precise ser expandido ou reduzido em suas dimensões. Isso poderia acontecer se, por implicações legais, fosse necessário manter dados das dez últimas contribuições e não somente das últimas cinco.

Muito provavelmente, caso cuidados adicionais não tivessem sido tomados, os algoritmos de manipulação dos campos poderiam sofrer impactos. Isso aconteceria também se o número de ocorrências viesse a diminuir. Sabemos que alguns artifícios poderiam até isolar esse tipo de impacto, mas considerando implementações rotineiras, podemos considerar esse exemplo como válido.

- **Impactos sobre a estabilidade lógica dos dados.** Considerando ainda o mesmo exemplo visto anteriormente, vamos analisar o impacto gerado sobre as estruturas de dados pelo aumento ou diminuição da quantidade de elementos do "item de repetição".
  Considere que tenha havido uma diminuição de 5 para 3 anos na necessidade de retenção dos dados dos valores de contribuição sindical. O que aconteceria com a estrutura da tabela? Teria de ser ajustada para "exclusão" das ocorrências número 4 e número 5 dentro do vetor. Isso implicaria, dependendo do gerenciador de bancos de dados utilizado, maior ou menor esforço, mas de qualquer modo haveria um reflexo direto na definição da estrutura da tabela.
  Ao mudarmos a estrutura da tabela, acabaríamos também por refletir alguns impactos, maiores ou menores, novamente, na estrutura lógica dos programas. Todos os programas que estivessem referenciando a ocorrência de número 4 ou número 5 de um vetor desse tipo, após a alteração da estrutura da tabela, teriam de passar por algum tipo de ajuste.
  Poderíamos nos alongar mais em exemplos que mostrassem a instabilidade imposta aos modelos lógicos de dados e aos processos, quando estruturas não normalizadas são utilizadas. Veremos mais alguns desses impactos quando mostrarmos as anomalias envolvidas em cada uma das fases de normalização de tabelas. Isso com certeza lhe dará mais elementos para avaliar essa questão.

2. Flexibilidade. Outra característica que pode vir a ser obtida com o processo de normalização das tabelas é a flexibilidade, tanto de processos como de utilização das estruturas de dados. Por flexibilidade entendemos a capacidade de adaptação a demandas diferenciadas, a expansão e redução, a omissão ou presença, etc.

Se, em uma estrutura de dados implementada, não tivermos a devida flexibilidade, estaremos impondo restrições de uso e tornando as estruturas de dados e de processos dependentes de limites rígidos. Isso fará com que se tornem, por conseguinte, instáveis o que nos trará implicações já abordadas no tópico anterior.

Vamos considerar ainda o exemplo utilizado na seção "Impactos sobre a Estabilidade Lógica das Aplicações". O fato de termos definido de modo não normalizado a estrutura de VALORES DE CONTRIBUIÇÃO SINDICAL faz com que tenhamos de definir um número específico de ocorrências a serem mantidas na tabela. Seja esse número 1, 2, 5, 10 ou 100, teremos sempre uma estrutura inflexível. Perceba que sempre o valor escolhido poderá ser excessivo ou insuficiente. Se optamos por uma única ocorrência para o vetor poderemos precisar de uma segunda e, então, não teremos como armazená-la. Se optarmos por 100 ocorrências no vetor poderemos precisar de 101 e também teremos o mesmo problema. Por outro lado, se tivermos 100 ocorrências previstas no vetor e um funcionário recém-admitido com somente uma contribuição, estaremos tendo uma estrutura que nos forçará a manter 99 ocorrências do vetor sem valores armazenados. O mesmo poderia ser verdadeiro se tivéssemos somente uma ocorrência no vetor e um funcionário fosse isento de contribuição sindical. Para esse caso, mesmo havendo somente uma única ocorrência prevista, ela nunca seria preenchida mas continuaria estando sempre presente na estrutura da tabela. Isso reflete uma estrutura inflexível, incapaz de se ajustar a variações, exceções, etc. Uma tabela que esteja na primeira forma normal não apresentará esse tipo de comportamento, sendo capaz de apresentar-se de modo bastante flexível num exemplo como o citado anteriormente.

3. Integridade. As estruturas de dados obtidas pelo processo de modelagem necessitam de recursos para que os dados a serem nelas armazenados possam ter "qualidade". A qualidade de um dado está vinculada a diversos fatores. Entre eles podemos citar atualidade, veracidade, fidelidade, acuracidade, integridade, etc.

A questão ligada à integridade dos dados tem sido bastante discutida quando ligada ao ambiente de bancos de dados relacionais. Isso muitas vezes tem levado a maioria das pessoas a pensar somente nos aspectos puramente lógicos da questão. Se uma referência de uma tabela a um valor de outra tabela é satisfeito, parece estar satisfeita a questão de integridade (pelo menos a integridade referencial). Entretanto, o próprio termo "integridade referencial" já foi introduzido com o complemento "referencial" para tornar distinto seu conceito do simples conceito de "integridade".

A "integridade" de modo genérico diz respeito à qualidade do dado. Se um dado sobre certo objeto aparecer mapeado em mais de um local de modo diferente, com valores instanciados de modo diferente, poderemos ter indícios de que não há integridade entre eles. Isso nem sempre será verdadeiro. Existirão casos onde esse dessincronismo será proposital, intencional, desejado ou até obrigatório. Entretanto, em tantos outros casos isso será realmente uma falta de integridade entre os dados existentes.

Esse tipo de anomalia é algo que a normalização das tabelas procurará coibir. As segunda e terceira formas normais nos induzirão a corrigir alguma anomalia que permita a existência de dados repetidos desnecessariamente e, assim, sujeitos à falta de integridade (não necessariamente "referencial").

Vamos imaginar o caso de uma tabela onde apareçam dados relativos às inscrições de alunos em um dado curso. Essa tabela poderia ter sido modelada, sem normalização, da seguinte maneira:

Tabela INSCRIÇÃO

Colunas:
Código do Aluno
Código da Disciplina
Nome do Curso
Telefone para Contato com o Aluno
Data da Matrícula

Essa tabela seria alimentada por um processo de matrículas onde cada aluno indicaria que disciplinas de curso pretende cursar durante o próximo semestre. Caso o aluno não fosse aceito para cursar essa disciplina por falta de vagas, pré-requisitos, ou por ter um índice de aproveitamento escolar muito baixo, ele deveria ser contactado para ser comunicado do cancelamento de sua inscrição.

Exemplo:

Código do Aluno	Código da Disciplina	Nome do Curso	Telefone Contato	Data Matrícula
012343	002	Direito	264-7205	25/11/95
012343	033	Administração	264-7205	20/12/95
094145	003	Direito	262-9981	09/11/95
094145	045	Psicologia	262-9981	09/11/95
094145	068	Psicologia	262-9981	10/11/95

Perceba que, por não estar devidamente normalizada, a tabela apresenta uma deficiência bastante óbvia. O telefone de contato com o aluno para comunicação de cancelamento de sua inscrição aparece de forma desnecessária repetido em cada uma das inscrições que ele tenha feito. E se o aluno mudar seu telefone? O que aconteceria se alguém atualizasse o telefone em somente uma de suas inscrições? Teríamos falta de integridade. Não falta de integridade referencial mas simplesmente falta de integridade na informação contida na tabela. Se consultássemos seu telefone em uma das inscrições teríamos um resultado, se consultássemos em outra teríamos outro.

4. Economia. O aspecto economia, quando se aborda o tema normalização, tem sido visto, basicamente, sob o prisma da economia de espaço para armazenamento. Mas seria esse o único motivo? Não temos visto o barateamento, cada vez maior, das mídias de armazenamento de massa? E se tivermos espaço de sobra? Mesmo assim deveríamos ter esse aspecto sob consideração?

A resposta é sim. Os custos de armazenamento, gerados pela redundância, pela desnormalização e por outras anomalias presentes na construção das tabelas não são, entretanto, os maiores custos existentes. Além do custo de armazenamento teremos também o custo de manipulação.

Esse custo representa todo e qualquer esforço, tempo, ou valor agregado ao fato de manipularmos volumes de dados maiores do que os efetivamente necessários. Tempos envolvidos em processamento, unidades de disco, controladoras, mídias para backup, local para armazenamento das mídias de backup e tantos outros elementos podem ter seus custos aumentados simplesmente porque nosso projeto não está otimizado. Por exemplo, preparar e manter rotinas de backup e recuperação de uma base de 2 gigabytes ao invés de uma base de somente 1 gigabyte envolverá com certeza custos diferentes.

A existência de redundâncias nos dados causará também efeitos sobre os processos. Os processos de atualização, de certo modo, também refletirão essa redundância, desde que se pretenda manter o sincronismo entre as cópias redundantes. Isso significa que para criar e manter esses processos redundantes teremos custos adicionais. Se o desenvolvimento de uma rotina de atualização de dados bancários de um funcionário envolve a análise, especificação, construção, teste e manutenção dos programas destinados a esse fim, imagine o que isso representará se tivermos essa informação redundada em cinco tabelas distintas, sem domínio sobre essa situação. Provavelmente teremos cinco processos semelhantes sendo concebidos, implementados, testados e mantidos por, até, no pior dos casos, cinco equipes distintas simultaneamente. Isso representará um custo bastante acima do ideal que seria trabalharmos com somente um processo.

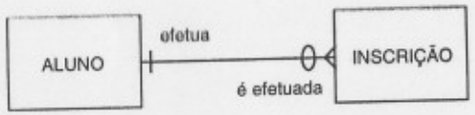
Outro custo, de certo modo intangível, está ligado ao fato de, por termos de executar mais processos e, por sua vez, processos mais demorados, termos também atraso no fornecimento da informação desejada. Essa

situação representará perda na imagem, no conceito, na eficiência e em outros pontos atrelados à qualidade do serviço prestado e, com certeza, recuperá-los futuramente envolverá custos bastante grandes.

5. Fidelidade ao Ambiente Observado. Esse aspecto, com certeza o mais subjetivo de todos os já apresentados até aqui, não deixa, entretanto, de ser extremamente importante. Como procuramos defender até o presente momento, a representação de um ambiente observado, através de um modelo, deve procurar retratar, independentemente do grau de abstração, as características observadas, assim como elas efetivamente são.

Caso tenhamos criado um modelo que não seja capaz de demonstrar, em sua totalidade, os elementos observados, deveríamos poder aplicar algum método que verificasse sua validade. Esse método é a normalização. Em suas segunda e terceira formas normais, teremos o processo de normalização, ajudando a definir elementos, até então despercebidos como elementos autônomos em nossos modelos.

Imaginemos o caso de um modelo onde dentre as entidades modeladas tivéssemos:



Para as entidades modeladas e, por conseguinte, para as tabelas propostas para a implementação, poderíamos ter os seguintes atributos:

Tabela ALUNO

Colunas:
Código do Aluno
Nome do Aluno
Ano de Admissão
Telefone de Contato

Tabela INSCRIÇÃO

Colunas:
Código do Aluno
Código da Disciplina
Nome da Disciplina
Código do Curso
Nome do Curso
Data da Matrícula

Exemplo:

ALUNO

Código do Aluno	Nome do Aluno	Ano de Admissão	Telefone para Contato
001	Paulo	1981	264-7205
002	Mário	1982	262-8303
003	Carlos	1982	222-1010
004	Nézio	1979	245-5678

INSCRIÇÃO

Código do Aluno	Código da Disciplina	Nome da Disciplina	Código do Curso	Nome do Curso	Data da Matrícula
002	004	Projeto II	001	Eng. Mecânica	23/01/96
002	009	Cálculo	001	Eng. Mecânica	23/01/96
003	004	Projeto II	001	Eng. Mecânica	31/10/95
003	007	Análise	002	Proc. Dados	01/11/95
003	010	Algoritmos	002	Proc. Dados	02/11/95

Perceba que também fica bastante óbvia a anomalia existente na tabela INSCRIÇÃO. A cada inscrição, para uma mesma disciplina (004 por exemplo), temos de repetir alguns dados da própria disciplina (o nome, neste exemplo) e também dados do curso ao qual pertence essa disciplina (001 — Eng. Mecânica).

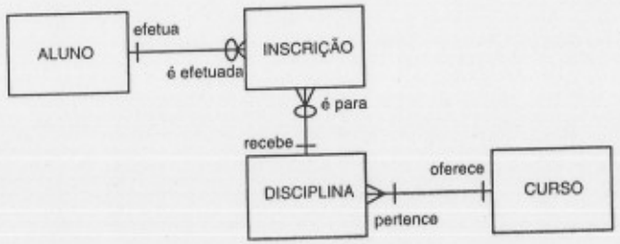
Além do aspecto desestruturação dos dados já visto, deveríamos ser capazes de perceber, nessa tabela, e no modelo concebido, que temos a ausência de duas entidades, realmente existentes no ambiente observado: DISCIPLINA e CURSO.

Se alguém faz uma inscrição, essa inscrição é, certamente, para uma disciplina. Não se pode fazer uma inscrição em "nada". E, por sua vez, uma disciplina, se assim assumirmos, deve pertencer sempre a um e só um curso. (Observação: poderíamos ter concebido um modelo onde uma disciplina pudesse pertencer a mais de um curso simultaneamente. Não o fizemos a fim de simplificar o exemplo apresentado.)

Perceba que pela descrição do próprio processo de inscrição estamos enumerando elementos distintos: ALUNO, INSCRIÇÃO, DISCIPLINA

e CURSO. Caso esses elementos não tenham aparecido espontan mente durante a modelagem, eles deverão ser explicitados pelo processo de normalização.

Ao final da normalização, na terceira forma normal, teríamos o seguinte modelo:



Com as seguintes tabelas:

Tabela: ALUNO

Colunas:
Código do Aluno
Nome do Aluno
Ano de Admissão na Universidade
Telefone de Contato

Tabela: DISCIPLINA

Colunas:
Código da Disciplina
Nome da Disciplina

Tabela: INSCRIÇÃO

Colunas:
Código do Aluno
Código da Disciplina
Data da Matrícula

Tabela: CURSO

Colunas:
Código do Curso
Nome do Curso

Esse deveria ser o modelo a ser implementado para que obtivéssemos estabilidade, flexibilidade, economia e fidelidade. Assim, podemos perceber que com a normalização obteremos o modelo ideal a ser implementado mesmo que tenhamos cometido omissões ou falhas durante o processo de modelagem conceitual.

Primeira Forma Normal (1FN)

A definição para a primeira forma normal, assim como para as demais, é bastante formal e rigorosa. Não poderia e nem deveria ser diferente. Como esses processos agem sobre estruturas relacionais e foram definidos por pessoas ligadas ao ambiente matemático, herdaram predominantemente o formalismo presente nessas áreas. Isso é bom, por certo lado, pois torna inequívoca sua interpretação. Entretanto, por outro lado, temos percebido que boa parte da dificuldade de entendimento, no que diz respeito a formas normais, advém justamente da dificuldade em "traduzir o formalismo para termos do dia-a-dia".

Com esse intuito, estaremos omitindo, deliberadamente, as definições formais relativas às formas normais, e recomendando que aqueles que as desejem conhecer, na íntegra, busquem a literatura recomendada ao final do livro. Usaremos em nossas definições "traduções simplificadas" para apresentar as formas normais.

Definição da Primeira Forma Normal

Diz-se que um modelo está na primeira forma normal se:

— Está integrado por tabelas.
— As linhas da tabela são unívocas.
— As linhas não contêm itens repetitivos.
— Os atributos são atômicos.
— Os atributos não contêm valores nulos.

Iremos discorrer sobre cada um dos aspectos apresentados, visando a conceituá-los de forma mais clara.

"... Está integrado por tabelas"

Estamos objetivando com as regras das formas normais obter modelos lógicos relacionais que contenham estruturas adequadamente validadas e verificadas. Se nosso objetivo é conceber um modelo relacional correto, o primeiro ponto a ser obedecido é que ele esteja definido em forma de TABELAS, ou seja, que respeite a característica de ser implementado através de LINHAS e COLUNAS unicamente. Já vimos anteriormente, quando falamos sobre os conceitos envolvidos no ambiente relacional, essas definições. Assim, se nosso modelo não puder ser visto através de linhas e colunas não teremos sequer iniciado o atendimento do primeiro requisito da 1FN, logo devemos buscar sua adequação.

"... As linhas da tabela são unívocas"

Também quando mostramos os conceitos envolvidos no ambiente relacional, falamos sobre a obtenção de identificadores únicos para uma tabela visando a assegurar a diferenciação entre duas linhas de uma tabela. Apresentamos a correlação entre TABELAS e RELAÇÕES e a justificativa para que seja proibida a existência de duas linha iguais. Assim, dentro do processo de normalização e obtenção da 1FN devemos definir obrigatoriamente o(s) identificador(es) unívoco(s), ou chave(s) primária(s) da tabela. Só assim asseguraremos a não existência de linhas iguais.

"... As linhas não contêm itens repetitivos"

Grupos repetitivos são estruturas de dados artificiais (vetores) criadas para facilitar o manuseio de ocorrências repetitivas de elementos de dados. Como uma RELAÇÃO é definida, obrigatoriamente, como um conjunto de domínios não necessariamente distintos, deveremos também na definição de nossas TABELAS apresentar os elementos repetitivos de modo isolado, dando a cada um uma denominação distinta, ou utilizar uma outra estrutura que os apresente associados ao conjunto inicial de dados não repetitivos.

Inicialmente, a totalidade dos gerenciadores de bancos de dados relacionais omitia qualquer recurso para definição e manipulação de estruturas do tipo vetor (itens de repetição). Hoje, alguns fornecedores de SGBDs, trabalhando sobre extensões do padrão SQL ANSI, podem oferecer alternativas para que se tenha uma visibilidade externa através de itens de repetição enquanto que internamente o SGBD continua atribuindo nomes distintos a cada uma das colunas, na execução de primitivas de manuseio. Apesar de possíveis, devem ser consideradas anomalias tais estruturas. Elas não seguem padrões SQL de mercado e logo podem vir a ser omitidas em futuras releases de produtos, o que poderá causar impactos sensíveis àqueles que as adotarem.

"... Os atributos são atômicos"

Estruturas de agrupamento de itens de dados tornaram-se bastante comuns junto a linguagens tais como o COBOL. Definir um item de dado do tipo DATA e sob ele hierarquizar subdefinições para DIA, MES e ANO é praticamente uma regra no ambiente de definição de dados do COBOL. Entretanto, esses agrupamentos e suas redefinições são estruturas não permitidas no ambiente relacional. Isso novamente tem correlação com a definição de RELAÇÃO. Alguém consegue conceber um domínio DATA subdividido em subdomínios DIA, MES e ANO sob o ponto de vista matemático? A resposta é não. Não há esse tipo de percepção no âmbito dos domínios.

Quando tratamos com estruturas relacionais, não temos mais o conceito de REGISTRO, aquele elemento que posicionalmente agregava um conjunto estável de CAMPOS em uma ordem predefinida. Assim, era possível criar máscaras que agrupassem campos contíguos redefinindo-os com outras características. Agora no ambiente relacional temos coleções de domínios que formam, é verdade, tabelas, mas sua ordem por definição não é assegurada.

Assim, poderemos ter em uma tabela, teoricamente, cada uma das linhas com suas colunas ordenadas em uma seqüência diferente. Isso faz com que não seja possível agrupar colunas contíguas, pois elas não mais existem sob esse prisma.

Exemplo:

No ambiente COBOL

nome	data		
	dia	mês	ano
Maria	31	12	1995
José	01	03	1994
Carlos	25	03	1996
Antônio	13	07	1995

No ambiente RELACIONAL

Relação = (nome, dia, mês, ano)

Maria	1995	31	12
01	03	1994	José
03	Carlos	1996	25
1995	07	Antônio	13

Perceba que no ambiente relacional temos em cada uma das linhas uma coleção de domínios que aparece em diferentes ordens. O importante é que cada uma das linhas contém o mesmo conjunto de domínios. Sua ordem, matematicamente falando, é indiferente. É verdade que, hoje, a grande maioria dos SGBDR não tem esse método de implementação de tabelas. Eles, ainda, respeitam um posicionamento das colunas de acordo com a definição das tabelas. Isso, entretanto, não faz com que a definição de não-obrigatoriedade posicional deixe de ser verdadeira. Essa é uma característica potencialmente existente em um SGBDR e, portanto, devemos respeitá-la durante nosso projeto.

"... Os atributos não contêm valores nulos"

A definição da 1FN impondo a não existência de valores nulos era, historicamente, uma restrição bastante clara. Seguindo-se a definição formal de uma relação não deveríamos, realmente, ter um domínio sem instanciação de valor. Seria o mesmo que dizer que teríamos uma relação formada pelos domínios ANIMAIS, CORES e FRUTAS e permitir a existência de uma relação:

R = {CÃO, AMARELO, BANANA; PRETO, GATO, MAÇÃ; AZUL, COELHO}

Perceba que a terceira tupla da relação R não apresenta os três domínios instanciados. Temos uma COR, um ANIMAL mas não temos uma FRUTA. Isso equivale a dizer que o domínio FRUTA tem um valor NULO associado na tupla ou, simplesmente, inexiste.

Logo, se uma tabela representasse fielmente uma relação, ela não deveria permitir a existência de valores NULOS. Acontece que, por motivos práticos, isso ocorre (até com uma certa frequência) em nosso dia-a-dia. Tratar todos os casos de colunas onde possam aparecer valores nulos, normalizando-os, pode ser uma tarefa pouco prática de efeitos operacionais indesejáveis.

Deve-se tomar cuidado, entretanto, ao analisar as colunas onde aparecem valores NULOS para identificar se estes são DESCONHECIDOS ou INAPLICÁVEIS. Valores NULOS originários de situações caracterizadas por DESCONHECIDOS são comuns, devem ser tratados naturalmente e não indicam qualquer anormalidade. Já valores NULOS originários de situações caracterizadas por INAPLICÁVEIS podem denotar irregularidades no modelo conceitual e devem ser analisados com maior cuidado. Caso sejam detectadas situações que mereçam a definição de tabelas distintas, isso deverá ser feito. Veremos mais a frente como tratar os modelos conceituais onde apareçam estruturas de Generalização-Especialização visando a criar tabelas únicas ou separadas para suas implementações.

A maior prova de que hoje os SGBDRs já aceitam com naturalidade a existência de valores NULOS nas tabelas normalizadas é a previsão, em nível de padrão SQL, de uma cláusula para que se defina a cada coluna se é "proibida a existência de valores nulos ou não". Caso não seja definida a "proibição", será assumida como default (padrão) a possibilidade de existência de valor nulo em uma coluna. Isso nos leva a considerar que:

> *Do ponto de vista prático atual, não há mais restrição quanto à existência de valores nulos em colunas de uma tabela para que ela esteja na primeira forma normal (1FN).*

Processo para obtenção da Primeira Forma Normal (1FN)

1. Definir as chaves candidatas e escolher a chave primária da tabela.
2. Transformar atributos compostos em atômicos.
3. Em cada tabela eliminar grupos repetitivos gerando novas linhas, uma para cada ocorrência de item repetitivo, mantendo os valores dos demais itens.

1. Definir as Chaves Candidatas e Escolher a Chave Primária (PK) da Tabela. Esse passo, que em um primeiro instante parece trivial, pode nos levar a uma série de considerações de ordem prática. Em princípio, a escolha da chave da tabela deve ser feita observando-se os atributos existentes na tabela e escolhendo-se um ou mais deles para que concatenados possam servir como meio de diferenciação entre uma linha e outra (função básica da chave primária).

Como o conceito de chave primária não tem mais amarração com o método de recuperação, teoricamente, poderemos utilizar tantos atributos (ou colunas) quantos forem necessários à formação da chave da tabela, mesmo que tenhamos que chegar à concatenação de todos os atributos de uma linha. Isso, é claro, apesar de conceitualmente correto, não traz resultados práticos muito agradáveis. Como o estabelecimento dos relacionamentos entre as tabelas será feito através de chaves estrangeiras, imagine o que aconteceria se tivéssemos uma tabela onde todas as suas 10 colunas servissem com formadoras de sua chave primária? Teríamos, na tabela a ser relacionada a essa, necessariamente repetidas todas as 10 colunas da tabela de origem. Isso não seria, teoricamente, prejudicial ou ruim, mas na prática pode trazer alguns impactos do ponto de vista de alocação física de espaços, manuseio de estruturas de índices, mudanças em valores ou formatos de colunas, etc.

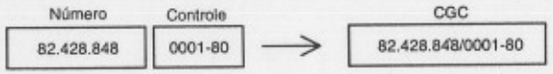
Em outra situação extrema, também poderemos perceber que nenhuma das colunas ou suas associações entre si nos levará à identificação de uma chave candidata. Isso poderá acontecer em função de diferentes motivos mas, certamente, você um dia se defrontará com essa situação.

O importante, em ambos os casos, é que fiquemos atentos ao processo de definição de chaves primárias. Caso seja percebida a inexistência de uma chave candidata ou a necessidade de concatenação de um número muito grande de atributos, então deverá ser definida uma nova coluna, dita "artificial", para que supra a demanda necessária de identificação. Essa coluna poderá ser uma coluna do tipo CONTADOR ou NÚMERO SEQÜENCIAL, ou mesmo algum outro artifício para composição com as colunas já existentes. Entretanto, esses recursos deverão ser utilizados com a máxima restrição e bom senso. Muitos projetistas acabam definindo muitas "colunas artificiais" sem necessidade. Como saber quando e onde utilizá-las? Somente a experiência dirá. Melhores ou piores projetos terão sempre esse diferencial.

2. Transformar Atributos Compostos em Atômicos. Esse passo é bastante simples e implica poucas decisões. Atributos compostos, que claramente tenham elementos distintos em sua estrutura, deverão ser separados. Por exemplo:

É certo que o número do telefone também poderia ser desmembrado em PREFIXO DA CENTRAL e MILHAR. Vamos simplificar essa estrutura deixando-a somente com esses dois atributos atômicos, por enquanto. O importante é perceber que PREFIXO DDD é um atributo distinto em relação a NÚMERO DO TELEFONE. Eles traduzem informações diferentes, têm características diferentes, regras de validação diferentes, domínios distintos, ou seja, são coisas diferentes e devem ser tratados separadamente.

Poderão surgir situações em que atributos compostos possam ser transformados em um único atributo atômico, pela perda da identidade de cada um dos elementos anteriores. Vejamos o exemplo a seguir. O atributo NÚMERO DO CGC e o atributo CONTROLE formadores do CGC poderão ser agregados e dar origem a somente um atributo chamado CGC. Teremos que, a partir desse instante, tratar sempre esse atributo como atômico. Caso desejemos recuperar todos os CGCs com números iguais a um certo valor teremos de, através de recursos próprios, e não os do SQL, efetuar algum tipo de verificação para recuperá-los, pois o NÚMERO não mais existirá autonomamente.



3. Em Cada Tabela Eliminar Grupos Repetitivos Gerando Novas Linhas, uma para Cada Ocorrência de Item Repetitivo, Mantendo os Valores dos Demais Itens. A execução dessa atividade, isoladamente, poderá nos parecer bastante estranha. Ao olharmos a estrutura tabular que temos antes e depois dessa atividade, seremos levados a acreditar que tínhamos antes uma estrutura bem melhor do que depois de sua execução. Isso se deve ao fato de, aparentemente, estarmos introduzindo um grau de redundância bastante grande na tabela normalizada. De fato, se parássemos ao final dessa atividade, não dando prosseguimento à segunda forma normal (2FN), este seria realmente o resultado obtido: a geração de um grau elevado de redundância em nossa tabela.

Veremos, mais a frente, durante a execução da normalização na segunda forma normal (2FN), que voltaremos a obter uma estrutura bastante otimizada, corrigindo, de certo modo, a redundância aparente introduzida nessa atividade.

Alguns autores, baseados no aspecto prático do processo de normalização, mudam a especificação dessa atividade, enunciando-a como:

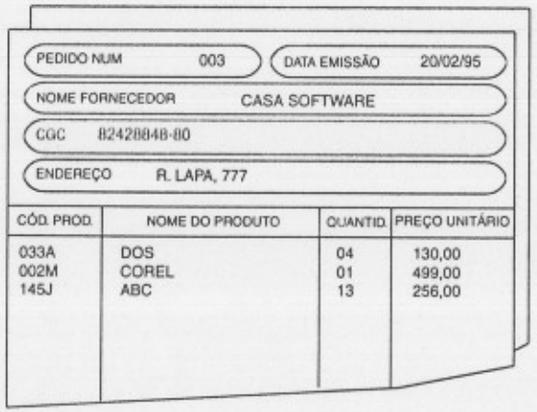
Enunciado alternativo para complementação da 1FN:

Para cada tabela, eliminar os grupos repetitivos gerando uma nova tabela, na qual cada um dos itens repetitivos dará origem a uma nova linha e na qual a chave primária será a concatenação da chave da tabela original com uma nova coluna que identifique de modo unívoco a linha.

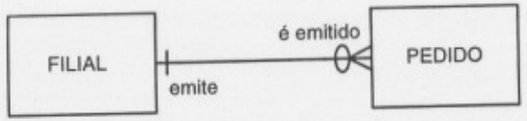
Esse enunciado realmente nos levará ao mesmo resultado da aplicação do enunciado original seguido da aplicação da segunda forma normal. Pode-se perceber que o que se pretende com ele é ser mais objetivo no processo de normalização, "chegando mais rapidamente ao resultado final". Apesar de esse enunciado não ser, fielmente, aderente ao processo de normalização, baseado nos conceitos matemáticos de relação, ele é, efetivamente, correto e, portanto, válido. Durante o exemplo que daremos a seguir, procuraremos apresentar o método convencional, mesmo que tenhamos que aguardar até a segunda forma normal para restabelecer a estrutura otimizada da tabela. Quando mostrarmos o exemplo de normalização na segunda forma normal, voltaremos a apresentar essa questão, relativa ao enunciado "alternativo" para a primeira forma normal e, então, você verá que os resultados obtidos realmente são idênticos.

Exemplo de normalização na 1FN

Vamos imaginar um processo de modelagem de uma empresa que é composta por várias filiais. Cada uma das filiais, periodicamente, efetua pedidos a seus fornecedores diretamente. Cada um dos pedidos pertence a somente um fornecedor, mas um pedido pode requisitar mais de um produto. O pedido utilizado pelas filiais tem o seguinte aspecto:



Durante a fase de análise de dados e elaboração do modelo [illegible], foram percebidas somente as seguintes entidades, atributos e relacionamentos:



Entidade: FILIAL
Atributos:

CGC
RAZÃO SOCIAL
TELEFONE
ENDEREÇO
BAIRRO
CEP

Entidade: PEDIDO
Atributos:

NÚMERO DO PEDIDO
DATA DE EMISSÃO
NOME DO FORNECEDOR
CGC DO FORNECEDOR
ENDEREÇO DO FORNECEDOR
PRODUTOS (ATÉ 10) CONTENDO :
- CÓDIGO DO PRODUTO
- NOME DO PRODUTO
- QUANTIDADE PEDIDA
- PREÇO UNITÁRIO

Perceba que durante o processo de modelagem somente foram identificadas duas entidades: FILIAL e PEDIDO. Você, com certeza, já teria percebido outras entidades distintas tais como PRODUTO, FORNECEDOR, etc. Entretanto, vamos partir de um modelo NÃO NORMALIZADO para, também, mostrar que o processo de normalização fará com que surjam, espontaneamente, as entidades ainda não modeladas.

O primeiro passo necessário para a normalização em 1FN é obter uma forma tabular de representação dos dados. Isso implica dispô-los em formato de linhas e colunas. Para facilitar esse entendimento, e também a visualização do processo de normalização, iremos montar uma tabela contendo linhas e colunas que servirão como exemplo. Esse procedimento não é um passo obrigatório durante a normalização. Ele serve somente como meio para facilitar a visualização do próprio processo.

O processo d[illegible]rmalização deverá ser feito para cada uma das tabelas existentes no modelo já obtido. Iremos tratar, entretanto, somente a tabela PEDIDO (que já sabemos de antemão não está normalizada). A tabela FILIAL poderá ser validada por você posteriormente.

Observação: Iremos incluir na tabela do exemplo outros pedidos executados para outros fornecedores, a fim de poder dar uma visão mais abrangente do possível conteúdo de uma tabela de PEDIDOS.

TABELA **PEDIDO** *NÃO-NORMALIZADA*

Núm. Ped.	Data Emiss.	Nome Fornecedor	CGC	Endereço	PRODUTOS			
					cód. prod.	nome prod.	qtde.	preço unit.
003	20-Jan	Casa Software	8242888-80	Lapa777	033A	DOS	04	130,00
					002M	COREL	01	499,00
					145J	ABC	13	256,00
004	27-Jan	Brasilsoftware	8047999-70	Itu 49	002M	COREL	02	450,00
					083P	ZAPT	10	85,00
005	27-Jan	Computerland	7932253-23	Feijó 122	033A	DOS	50	110,00
					145J	ABC	50	110,00
006	14-Mar	Brasilsoftware	8047999-70	Itu 49	029K	WIN	15	200,00
					083P	ZAPT	10	87,00

Essa é a primeira representação possível, em formato tabular, da entidade PEDIDO e seus atributos (agora colunas). Perceba que transformamos, em princípio, nossas estruturas de dados em um formato de linhas e colunas. Temos uma tabela com 4 linhas e 6 colunas. Note que o pedido 003 está representado, por enquanto, por só uma linha. Note também que a coluna PRODUTOS é um item de repetição onde podem aparecer até 10 produtos de um mesmo pedido. Podemos perceber, claramente, que essa tabela ainda não está na primeira forma normal. Por quê?

Porque para ter uma tabela na primeira forma normal devemos:

1. Acabar com os itens de repetição.
2. Transformar os atributos em atributos atômicos.
3. Definir uma chave para que tenhamos unicidade nas linhas da tabela.

Vamos analisar cada um desses passos e executá-los sobre a tabela anterior:

1. Acabar com Itens de Repetição. Seguindo as explicações dadas anteriormente, deveremos tomar cada um dos itens de repetição e com eles dar origem a novas linhas da tabela onde o conteúdo das demais colunas será o mesmo da linha original. Fazendo isso, obteremos a seguinte tabela:

Núm. Ped.	Data Emiss.	Nome Fornecedor	CGC	Endereço	Cód. Prod.	Nome Prod.	Qtde.	Preço Unit.
003	20-Jan	Casa Software	8242888-80	Lapa 777	033A	DOS	04	130,00
003	20-Jan	Casa Software	8242888-80	Lapa 777	002M	COREL	01	499,00
003	20-Jan	Casa Software	8242888-80	Lapa 777	145J	ABC	13	256,00
004	27-Jan	Brasilsoftware	8047999-70	Itu 49	002M	COREL	02	450,00
004	27-Jan	Brasilsoftware	8047999-70	Itu 49	083P	ZAPT	10	85,00
005	27-Jan	Computerland	7932253-23	Feijó 122	033A	DOS	50	110,00
005	27-Jan	Computerland	7932253-23	Feijó 122	145J	ABC	50	110,00
006	14-Mar	Brasilsoftware	8047999-70	Itu 49	029K	WIN	15	200,00
006	14-Mar	Brasilsoftware	8047999-70	Itu 49	083P	ZAPT	10	87,00

2. Transformar os Atributos em Atributos Atômicos. Analisando os atributos existentes podemos identificar alguns que podem merecer cuidados adicionais. Definiremos que:

- O atributo DATA DE EMISSÃO será tratado de forma atômica através de uma coluna definida na tabela como pertencente ao domínio DATE. Esse é um tipo de dado especial em que toda a data é manipulada como um item atômico.
- O atributo CGC será tratado, também, de forma atômica, não sendo separado em NÚMERO e CONTROLE.
- O código do produto será tratado como atômico, apesar de sua estrutura conter um código alfanumérico.
- O endereço será desmembrado em RUA e NÚMERO DO IMÓVEL por se tratar de dados com naturezas completamente distintas, representando características completamente diferentes.

Caso desejássemos separar, por exemplo, o CGC em duas colunas distintas NÚMERO e CONTROLE, bastaria representá-las de modo autônomo e prosseguir com as próximas atividades.

Adotando-se as estratégias citadas, acabaremos por obter a seguinte tabela:

Núm. Ped.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.	Cód. Prod.	Nome Prod.	Qtde.	Preço Unit.
003	20-Jan	Casa Software	8242888-80	Lapa	777	033A	DOS	04	130,00
003	20-Jan	Casa Software	8242888-80	Lapa	777	002M	COREL	01	499,00
003	20-Jan	Casa Software	8242888-80	Lapa	777	145J	ABC	13	256,00
004	27-Jan	Brasilsoftware	8047999-70	Itu	49	002M	COREL	02	450,00
004	27-Jan	Brasilsoftware	8047999-70	Itu	49	083P	ZAPT	10	85,00
005	27-Jan	Computerland	7932253-23	Feijó	122	033A	DOS	50	110,00
005	27-Jan	Computerland	7932253-23	Feijó	122	145J	ABC	50	110,00
006	14-Mar	Brasilsoftware	8047999-70	Itu	49	029K	WIN	15	200,00
006	14-Mar	Brasilsoftware	8047999-70	Itu	49	083P	ZAPT	10	87,00

3. Definir uma Chave para que Tenhamos Unicidade nas Linhas da Tabela. Para a definição da chave da tabela, deveremos observar os atributos da tabela e procurar identificar aqueles que sozinhos, ou associados a outros atributos, garantam a não repetição de valores em mais de uma linha. Normalmente, esses atributos estarão associados a domínios definidos para CÓDIGOS ou NÚMEROS, entretanto, esta não é uma exigência de caráter teórico e sim prático.

No caso da tabela PEDIDO, um atributo que nos chama a atenção, em um primeiro instante, é o próprio NÚMERO DO PEDIDO. Se os pedidos já têm, originalmente, um número, é grande a probabilidade de que ele sirva como meio para diferenciar linhas dessa tabela. Para verificar se esse atributo serve como chave, deveríamos aplicar algumas indagações sobre sua aplicabilidade:

- Um número de pedido pode se repetir em uma mesma filial?
- Um número de pedido pode se repetir em diferentes filiais?

Por definição, em nosso ambiente observado, foram levantadas essas informações e a resposta em ambos os casos foi NÃO. Caso a primeira pergunta tivesse resposta positiva, teríamos de optar por uma outra chave identificadora para a tabela PEDIDO diferenciando de algum modo, talvez com a agregação de algum outro atributo, dois pedidos que tivessem o mesmo número em uma mesma filial. Caso a segunda pergunta tivesse resposta positiva, teríamos que, ao menos, agregar como atributo de identificação do pedido, o atributo de identificação da filial. Isso tornaria dois pedidos de número igual, porém de filiais diferentes, distintos entre si. Caso as duas perguntas tivessem resposta positiva, teríamos que agregar as estratégias utilizadas em cada um dos casos isoladamente e, então, obter uma solução global que atendesse às duas situações.

Já sabemos, até o presente momento, que o atributo NÚMERO DO PEDIDO não se repete para dois pedidos distintos. Mas seria isso sufi-

ciente? A resposta é NÃO. Precisamos tornar distintas duas linhas da tabela PEDIDO e isso envolve analisar toda a linha. Se observarmos, logo de início, as linhas 1, 2 e 3 da tabela, perceberemos que todas têm o NÚMERO DO PEDIDO igual a 003. Também nas linhas 4 e 5 temos a repetição do NÚMERO DO PEDIDO igual a 004 e assim por diante. Fica claro perceber que somente com o atributo NÚMERO DO PEDIDO não conseguiremos identificar de modo distinto uma linha da tabela, pois dado o número de pedido igual a 005 poderemos identificar mais de uma linha simultaneamente.

Isso nos indica que, ou deveremos buscar outro identificador para a tabela, ou deveremos associar mais algum atributo ao atributo NÚMERO DO PEDIDO. Se analisarmos cada um dos atributos da tabela isoladamente teremos:

Dado o Atributo ...	Poderemos Identificar Quantas Linhas na Tabela?
NÚMERO DO PEDIDO	Várias (linha 1, 2 e 3 com 003)
DATA DE EMISSÃO	Várias (linhas 4, 5, 6 e 7 com 27-Jan)
NOME DO FORNECEDOR	Várias (linhas 4, 5, 8 e 9 com Brasilsoftware)
CGC DO FORNECEDOR	Várias (linhas 6 e 7 com 7932253-23)
RUA DO FORNECEDOR	Várias (linhas 4, 5, 8 e 9 com Rua Itu)
NÚMERO DO IMÓVEL	Várias (linhas 1, 2 e 3 com 777)
CÓDIGO DO PRODUTO	Várias (linhas 1 e 6 com 033A)
NOME DO PRODUTO	Várias (linhas 2 e 4 com COREL)
QUANTIDADE PEDIDA	Várias (linhas 5 e 9 com 10)
PREÇO UNITÁRIO	Várias (linhas 6 e 7 com 110,00)

Isso nos leva a considerar que, isoladamente, nenhum dos atributos da tabela é um CANDIDATO A CHAVE. Teremos, então, que partir para associações entre mais de um atributo.

Como, para chegarmos à tabela atual, fizemos uma "explosão" dos itens de repetição relativos a PRODUTOS, é bastante provável, senão quase certo, que encontraremos nos atributos relativos a PRODUTOS o atributo que, associado ao NÚMERO DO PEDIDO, nos levará a um identificador para a tabela.

Deveremos, portanto, analisar se, dado o NÚMERO DO PEDIDO e mais um atributo do PRODUTO, obteremos um chave unívoca. Vejamos:

Dados os Atributos ...	Obteremos Quantas Linhas na Tabela?
NÚMERO DO PEDIDO + CÓDIGO DO PRODUTO	Somente 1 (linha 5 com 004 e 083P)
NÚMERO DO PEDIDO + NOME DO PRODUTO	Somente 1 (linha 9 com 006 e ZAPT)
NÚMERO DO PEDIDO + QUANTIDADE PEDIDA	Várias (linhas 6 e 7 com 005 e 50)
NÚMERO DO PEDIDO + PREÇO UNITÁRIO	Várias (linhas 6 e 7 com 005 e 110,00)

Perceba que as combinações **NÚMERO DO PEDIDO + CÓDIGO DO PRODUTO** e também **NÚMERO DO PEDIDO + NOME DO PRODUTO** são chaves candidatas da tabela, pois não definem potencialmente mais de uma linha. Dissemos que são potencialmente candidatas a chaves porque, simplesmente, essa é a regra atual do ambiente modelado. Em outra situação poderiam não ser. Em nosso exemplo, estamos assumindo que em um mesmo pedido não aparecerão duas, ou mais vezes, repetidos o mesmo código de um produto. Se alguém desejar pedir para o mesmo fornecedor duas vezes o mesmo produto em um mesmo pedido, bastará mudar a quantidade pedida na primeira especificação ao invés de criar uma outra linha dentro do pedido. Isso poderia ser diferente em outra empresa e, então, também, somente esses dois atributos associados não mais seriam suficientes.

Em nosso caso, como temos duas chaves candidatas, escolheremos como chave da tabela PEDIDO a associação **NÚMERO DO PEDIDO + CÓDIGO DO PRODUTO**, pois nos parece mais natural trabalhar, internamente no sistema, com CÓDIGOS do que com NOMES. Isso não significa que o usuário não possa vir a ter outra visibilidade, associada aos NOMES DE PRODUTOS, em sua aplicação.

Com isso, obteremos a seguinte tabela:

Núm. Ped.	Cód. Prod.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.	Nome Prod.	Qtde.	Preço Unit.
003	033A	20-Jan	Casa Software	8242888-80	Lapa	777	DOS	04	130,00
003	002M	20-Jan	Casa Software	8242888-80	Lapa	777	COREL	01	499,00
003	145J	20-Jan	Casa Software	8242888-80	Lapa	777	ABC	13	256,00
004	002M	27-Jan	Brasilsoftware	8047999-70	Itu	49	COREL	02	450,00
004	083P	27-Jan	Brasilsoftware	8047999-70	Itu	49	ZAPT	10	85,00
005	033A	27-Jan	Computerland	7932253-23	Feijó	122	DOS	50	110,00
005	145J	27-Jan	Computerland	7932253-23	Feijó	122	ABC	50	110,00
006	029K	14-Mar	Brasilsoftware	8047999-70	Itu	49	WIN	15	200,00
006	083P	14-Mar	Brasilsoftware	8047999-70	Itu	49	ZAPT	10	87,00

Para fins de melhor visualização, colocamos as colunas pertencentes à chave identificadora da tabela juntas como as primeiras colunas da tabela. Isso não é, necessariamente, um requisito formal. Perceba, também, que utilizamos os nomes das colunas sublinhados para denotar que essas colunas são as chaves da tabela. Alguns outros autores utilizam outras notações possíveis, tais como:

- Delimitar os nomes das colunas pelos símbolos "<" e ">" como, por exemplo:

<Núm Ped>	<Cód. Prod.>	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.	Nome Prod.	Qtde.	Preço Unit.

- Prefixar o nome das colunas pelo símbolo "#":

#Núm. Ped.	#Cód. Prod.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.	Nome Prod.	Qtde.	Preço Unit.

Observação: Alguns autores utilizam o símbolo "#" simplesmente como substituto para NÚMERO ou CÓDIGO, sem associá-lo ao conceito de chave identificadora. Isso poderá confundi-lo em alguns instantes. Procure identificar bem qual é a notação utilizada em cada uma das ferramentas CASE, técnicas de diagramação ou representações existentes nos livros. Exemplo:

#Ped	#Prod.	Data Emiss.	Nome Fornecedor	CGC	Rua	#Imóv.	Nome Prod.	Qtde.	Preço Unit.

- Escrever ao lado do nome da coluna a sigla PK (*Primary Key*):

Núm. Ped. (PK)	Cód. Prod. (PK)	Data Emiss.	Nome Fornecedor	CGC	Rua	#Imóv.	Nome Prod.	Qtde.	Preço Unit.

Tendo atingido esse ponto, temos a tabela PEDIDO normalizada em sua primeira forma normal (1FN). Isso, com certeza, não é o suficiente. Ainda temos, nessa situação, uma série de anomalias na tabela gerada. Para sanar essas anomalias, deveremos buscar a segunda formal normal (2FN). Vejamos quais são os problemas existentes nessa forma normal.

Anomalias existentes em uma tabela na Primeira Forma Normal (1FN)

Iremos abordar as anomalias existentes na primeira forma normal dividindo-as em grupos. Esses grupos estarão sendo definidos pelas operações que poderíamos desejar efetuar sobre a tabela. Se para essa operação existirem restrições ou deficiências, então teremos caracterizado uma anomalia.

- INSERÇÃO:

Só é possível incluir um novo fornecedor a partir de um pedido.

Se nosso sistema fosse, única e exclusivamente, baseado na tabela apresentada até o momento, não poderíamos cadastrar um novo fornecedor em nossa tabela, a menos que surgisse um pedido para ele. Assim, se desejássemos incluir o fornecedor SoftwareTech, bem como seus dados, teríamos que aguardar até que um pedido fosse gerado para ele. Fica claro que deveríamos ter, de alguma maneira, uma tabela distinta só para o armazenamento dos fornecedores. Essa tabela surgirá a partir da aplicação da segunda e terceira formas normais.

Se incluirmos um pedido novo com CGC DO FORNECEDOR = 8242888-80 e com o NOME DO FORNECEDOR = "Casa dos Softwares", qual será, no futuro, a denominação válida para esse fornecedor? "Casa Software" ou "Casa dos Softwares"?

Como a inclusão dos dados do fornecedor está ligada a um pedido, nada impede que dois pedidos diferentes contenham dados diferentes para um mesmo fornecedor. Isso poderá acontecer com qualquer um dos atributos do fornecedor: nome, telefone, endereço, etc. Parece também claro que isso nos aponta no caminho de uma tabela exclusiva para conter os dados do fornecedor. Uma tabela onde uma vez cadastrados os dados de um fornecedor não mais precisássemos repetir essa operação. A tabela surgirá da aplicação da segunda e terceira formas normais.

- ELIMINAÇÃO:

Se eliminarmos o pedido 005, perderemos toda a informação sobre o fornecedor "Computerland".

Nesse caso, podemos perceber que o fato de um pedido conter, em sua estrutura, os dados do fornecedor, vinculados diretamente a sua existência, pode nos levar a, simplesmente, perder esses dados quando um pedido for excluído. Em nosso exemplo, como o único pedido onde os dados do fornecedor "Computerland" aparecem é o pedido 005, sua exclusão fará com que também os dados do fornecedor se percam.

- ATUALIZAÇÃO:

Se a data do pedido 003 tiver de ser alterada de 20-Jan. para 21-Jan., teremos que efetuar essa operação sobre várias linhas da tabela.

O fato de atributos pertencentes, única e exclusivamente ao pedido, estarem associados a linhas da tabela que representam cada um dos produtos do pedido, faz com que haja redundância de dados. Essa redundância, como já dissemos, nos levará à necessidade de redundância de processos, ou repetições de processos sobre mais elementos do que os essencialmente necessários.

Se algum dado do fornecedor "Brasilsoftware" mudar, teremos que atualizar diversas linhas da tabela, inclusive em diferentes pedidos.

Caso algum dos dados de um fornecedor venha a mudar, será necessário processar a alteração em cada uma das linhas de cada um dos pedidos pertencentes a esse fornecedor. Em nosso exemplo, se o CGC, telefone, endereço, ou qualquer outro dado do fornecedor "Brasilsoftware" tivesse de ser alterado, teríamos que efetuar essa operação sobre as linhas 4 e 5 (do pedido 004) e também nas linhas 8 e 9 (do pedido 006).

Segunda Forma Normal (2FN)

Como vimos até agora, aplicar as regras de obtenção da primeira forma normal sobre uma tabela poderá ainda não ser suficiente para a geração de uma estrutura sem anomalias. O intuito da segunda forma normal (2FN) é avançar, ainda mais, na direção de uma estrutura de dados que não contenha anomalias. Poderão existir casos onde somente aplicando-se as regras da primeira forma normal já venhamos a obter uma estrutura acabada. O que isso significa? Que, na verdade, nossa tabela já está normalizada, naturalmente, nas segunda e terceira formas normais. Para constatar isso, teremos de tentar aplicar as regras da segunda e terceira formas normais sobre a tabela. Se verificarmos que não existem as características indicadas pela segunda e terceira formas normais na tabela, então teremos comprovado sua normalização natural. O mesmo caso será verdadeiro em relação à segunda forma normal. Muitas vezes bastará que tenhamos aplicado as regras da segunda forma normal sobre uma tabela para que ela também já esteja normalizada na terceira forma normal.

Definição da Segunda Forma Normal

Uma tabela está na segunda forma normal se está na 1FN e cada uma das colunas não pertencentes à chave primária não for dependente parcialmente dessa chave.

Vejamos em detalhes a interpretação do enunciado:

"Está na primeira forma normal ..."

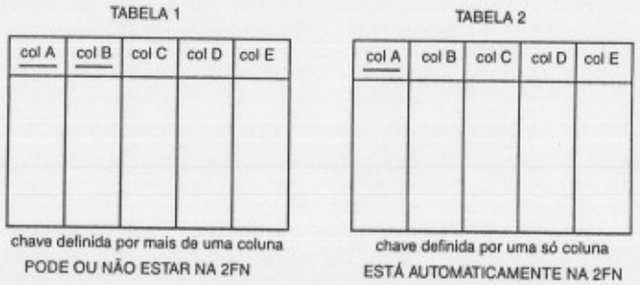
Para que se aplique a regra da segunda forma normal sobre uma tabela deve-se garantir, primeiramente, que ela já esteja na primeira forma normal. Caso isso não seja feito, não haverá sentido em se aplicar a 2FN.

"... cada uma das colunas não pertencentes à chave primária ..."

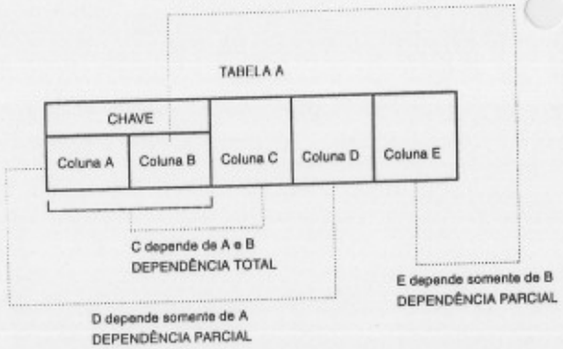
Esse enunciado deixa claro que ao analisar uma tabela, visando a verificar a necessidade de normalização na segunda forma normal deveremos "excluir" de nossa análise as colunas formadoras da chave primária dessa tabela. Isso é válido tanto para tabelas com somente uma coluna definindo a chave primária como para tabelas onde a associação de diversas colunas é necessária para a definição da chave.

"... não for dependente parcialmente dessa chave."

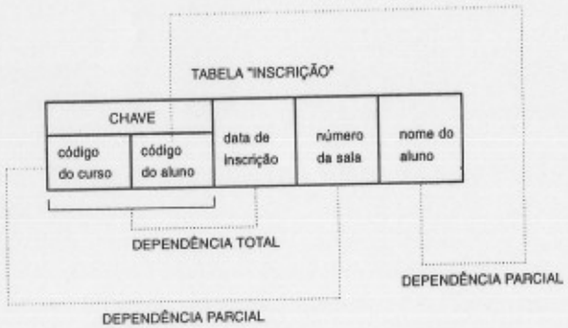
A dependência parcial de uma chave só será possível se a chave primária da tabela for definida por mais de uma coluna. Caso tenhamos somente uma coluna definindo a chave primária, essa tabela já estará, automaticamente, na segunda forma normal.



Mas o que significa exatamente dependência parcial da chave? Dizemos que uma coluna depende parcialmente da chave se, para que seu valor seja determinado não necessitarmos conhecer a chave como um todo, mas sim somente um ou alguns de seus valores. Assim, cada coluna de uma tabela que não pertença à chave e que possa ter seu valor determinado por um "pedaço" da chave é dita como "dependente parcialmente da chave".



Nesse exemplo, perceba que o valor da coluna E só depende do valor da coluna B para ser conhecido, logo depende parcialmente da chave da TABELA A. Já o valor da coluna C depende completamente da chave. Poderíamos ter um exemplo concreto que representasse essa situação do seguinte modo:



Imagine uma tabela de INSCRIÇÕES de alunos em determinados cursos. Ela já está na primeira forma normal mas ainda não na segunda forma normal. Temos claramente colunas que têm seus valores dependentes de somente parte da chave. No caso do NOME DO ALUNO, podemos perceber que conhecendo o CÓDIGO DO ALUNO poderemos determinar seu nome mesmo que não tenhamos conhecimento do CÓDIGO DO CURSO onde se inscreveu. O mesmo é válido para o NÚMERO DA SALA onde o curso será ministrado. Desde que conheçamos o CÓDIGO DO CURSO onde o aluno se inscreveu, poderemos determinar o NÚMERO DA SALA onde ele será ministrado mesmo que não saibamos o CÓDIGO DO ALUNO que está fazendo a INSCRIÇÃO. Essas duas situações denotam dependência parcial da chave para as colunas NÚMERO DA SALA e NOME DO ALUNO.

Por outro lado, para que possamos determinar a DATA DE INSCRIÇÃO deveremos conhecer tanto o CÓDIGO DO ALUNO como o CÓDIGO DO CURSO. Isso porque se só tivermos como informação o CÓDIGO DO ALUNO poderemos obter diversas DATAS DE INSCRIÇÃO, uma para cada curso onde ele se inscreveu. Do mesmo modo se só conhecermos o CÓDIGO DO CURSO poderemos obter diversas DATAS DE INSCRIÇÃO, uma para cada aluno que tenha se inscrito nesse curso. Essa situação denota a dependência total da chave para a coluna DATA DE INSCRIÇÃO.

Processo para obtenção da Segunda Forma Normal (2FN)

1. Identificar as colunas que não participam da chave primária da tabela.
2. Para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave.
3. Para as colunas dependentes parcialmente da chave:
   - Criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) da chave primária original que determinou(aram) o valor da coluna analisada.
   - Excluir da tabela original as colunas dependentes parcialmente da chave.

1. Identificar as colunas que não participam da chave primária da tabela.

O único cuidado a ser tomado nesse passo é identificar a notação utilizada para representar a tabela e sua chave. Nem sempre as colunas-chave estarão agrupadas no início da tabela ou se estiverem nem sempre utilizarão uma notação conhecida. A partir da identificação das colunas que formam a chave, bastará ignorá-las durante o processo de análise e o que restará serão as colunas que nos interessam.

2. Para cada uma das colunas identificadas, analisar se seu valor é determinado por parte, ou pela totalidade da chave.

A identificação da dependência parcial ou total da chave deverá ser feita coluna a coluna mesmo que isso pareça um processo demorado para aquelas tabelas que contenham dezenas de colunas. Essa análise poderá ser acelerada com a experiência de cada um no domínio que estiver sendo modelado. Para alguns será mais fácil identificar se uma coluna é dependente parcialmente da chave ou não. Já para outros, talvez seja necessário

algum apoio externo para essa identificação. De um modo ou d[illegible]tro, o fato importante é que isso deve ser feito e a hora é esta. Nem antes nem depois.

Para identificar a dependência de uma coluna em relação à chave deve-se indagar:

- Para que seu valor seja determinado, quais são as colunas da chave que devem ser conhecidas?

ou

- Qual seria o menor "pedaço da chave" que me permitiria obter um valor para a coluna analisada?

Se para alguma dessas perguntas for possível identificar um "pedaço" de chave que determine o valor de alguma coluna, teremos caracterizado uma tabela que ainda não está na segunda forma normal.

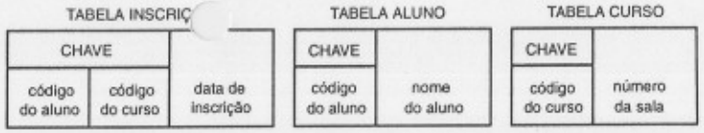
Uma boa estratégia é utilizar-se de exemplos com valores reais. Pense em valores instanciados nas colunas e imagine como eles estão interligados. Como uns determinam os outros.

3. Para as colunas dependente parcialmente da chave, criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) da chave primária original que determinou(aram) o valor da coluna analisada e excluir da tabela original as colunas dependentes parcialmente da chave.

Essa é a fase ativa desse processo de normalização. Até agora somente tentávamos identificar se a tabela precisaria ser normalizada ou não. Agora estaremos agindo sobre a tabela original, desmembrando-a em novas tabelas.

O processo de normalização da segunda forma normal dará origem a novas tabelas, fazendo com que objetos não observados distintamente e colocados juntos em uma mesma tabela venham a ser separados. Esse processo "garante" uma maior fidelidade ao processo de normalização. É claro que essa fidelidade também estará sendo fortemente dependente da correta identificação das colunas parcialmente dependentes da chave. Poderão vir a existir situações onde a compreensão equivocada de alguns elementos faça com que colunas parcialmente dependentes da chave continuem associadas à totalidade da chave simplesmente porque não as identificamos corretamente.

Aplicando-se essa regra sobre a estrutura da tabela INSCRIÇÃO, vista no exemplo das dependências parciais, passaríamos a ter as seguintes tabelas após a normalização na segunda forma normal:



Perceba que já passamos a ter três tabelas distintas ao invés de somente uma. Surgiram, pelo processo de normalização, as tabelas ALUNO e CURSO. Muito provavelmente elas deveriam ter sido percebidas já no instante da construção do modelo, mas, mesmo que não fossem, passariam a ser explicitadas por esse processo. Como dissemos, essa é uma das finalidades básicas da normalização.

Vamos voltar ao nosso exemplo completo em que apresentávamos uma tabela de PEDIDOS não normalizada. No passo anterior já transformamos a estrutura inicial dessa tabela em uma estrutura normalizada na primeira forma normal. O resultado que obtivemos foi o seguinte:

Núm. Ped.	Cód. Prod.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.	Nome Prod.	Qtde.	Preço Unit.
003	033A	20-Jan	Casa Software	8242888-80	Lapa	777	DOS	04	130,00
003	002M	20-Jan	Casa Software	8242888-80	Lapa	777	COREL	01	499,00
003	145J	20-Jan	Casa Software	8242888-80	Lapa	777	ABC	13	256,00
004	002M	27-Jan	Brasilsoftware	8047999-70	Itu	49	COREL	02	450,00
004	083P	27-Jan	Brasilsoftware	8047999-70	Itu	49	ZAPT	10	85,00
005	033A	27-Jan	Computerland	7932253-23	Feijó	122	DOS	50	110,00
005	145J	27-Jan	Computerland	7932253-23	Feijó	122	ABC	50	110,00
006	029K	14-Mar	Brasilsoftware	8047999-70	Itu	49	WIN	15	200,00
006	083P	14-Mar	Brasilsoftware	8047999-70	Itu	49	ZAPT	10	87,00

Aplicando-se sobre essa tabela a regra da segunda forma normal, deveremos:

"Analisar as colunas não chave..."

Essas colunas são: DATA EMISSÃO, NOME FORNECEDOR, CGC, RUA, NÚMERO IMÓVEL, NOME PRODUTO, QUANTIDADE, PREÇO UNITÁRIO.

"identificando as dependências parciais..."

Para cada uma das colunas descritas anteriormente, podemos identificar as seguintes dependências:

Coluna	Dependência	Chave
DATA EMISSÃO	PARCIAL	NÚMERO PEDIDO
NOME FORNECEDOR	PARCIAL	NÚMERO PEDIDO
CGC	PARCIAL	NÚMERO PEDIDO
RUA	PARCIAL	NÚMERO PEDIDO
NÚMERO IMÓVEL	PARCIAL	NÚMERO PEDIDO
NOME PRODUTO	PARCIAL	CÓDIGO PRODUTO
QUANTIDADE	TOTAL	NÚMERO PEDIDO + CÓDIGO PRODUTO
PREÇO UNITÁRIO	TOTAL	NÚMERO PEDIDO + CÓDIGO PRODUTO

Não é muito difícil perceber que se mudarmos somente o NÚMERO PEDIDO, independentemente do CÓDIGO PRODUTO, teremos determinado outros valores para DATA EMISSÃO, NOME FORNECEDOR, CGC, RUA e NÚMERO IMÓVEL. Estes são atributos que participam como um "cabeçalho" do pedido. Cada diferente PEDIDO tem esses valores determinados por seu NÚMERO PEDIDO.

Para a coluna NOME PRODUTO estabelecemos que, em nosso ambiente modelado, a regra existente é "um mesmo CÓDIGO PRODUTO" determina sempre um mesmo NOME PRODUTO, independentemente do PEDIDO. Isso porque, em nosso modelo, o CÓDIGO PRODUTO é algo existente no domínio do próprio cliente. Algo como um "cadastro de seus próprios produtos, com sua notação própria". Poderíamos ter uma outra realidade em outro exemplo. Poderiam existir casos onde cada diferente PEDIDO a um diferente fornecedor utilizasse os códigos do próprio fornecedor. Aí então teríamos que estabelecer outras dependências diferentes das citadas anteriormente.

Com relação à coluna QUANTIDADE, fica claro perceber que para se conhecer "quantas unidades" foram pedidas a um fornecedor é necessário saber "qual pedido" e "qual produto". Se só informarmos o NÚMERO PEDIDO teremos várias quantidades, uma para cada produto desse pedido. Se só informarmos o CÓDIGO PRODUTO teremos várias quantidades, uma para cada pedido onde apareça esse produto. Isso denota dependência total.

Para a coluna PREÇO UNITÁRIO, novamente, deveremos estabelecer qual é a regra válida para o ambiente que está sendo modelado. Para simplificar nosso exemplo, assumiremos que esse valor é uma cotação de preço do produto feita pelo fornecedor para um pedido especial e que não temos em "nosso cadastro" o preço associado permanentemente ao produto. Se tivéssemos o preço associado permanentemente ao produto deveríamos dizer que a dependência da coluna PREÇO UNITÁRIO seria parcial para a chave CÓDIGO PRODUTO.

"...criando novas tabelas com as colunas dependentes parcialmente das chaves, e excluindo essas colunas da tabela original."

Aplicando essa regra, obteremos as seguintes tabelas:

TABELA ORIGINAL
(colunas dependentes totalmente da chave)

Núm. Ped.	Cód. Prod.	Qtde.	Preço 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

TABELA OBTIDA POR DEPENDÊNCIA PARCIAL
(chave "NÚMERO PEDIDO")

Núm. Ped.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.
003	20-Jan	Casa Software	8242888-80	Lapa	777
004	27-Jan	Brasilsoftware	8047999-70	Itu	49
005	27-Jan	Computerland	7932253-23	Feijó	122
006	14-Mar	Brasilsoftware	8047999-70	Itu	49

TABELA OBTIDA POR DEPENDÊNCIA PARCIAL
(chave "CÓDIGO PRODUTO")

Cód. Prod.	Nome Prod.
033A	DOS
002M	COREL
145J	ABC
083P	ZAPT
029K	WIN

Anomalias existentes em uma tabela na Segunda Forma Normal (2F

Já vimos na primeira forma normal que, mesmo após realizado o processo de normalização, continuavam a existir anomalias na estrutura de dados obtida. Na segunda forma normal poderemos identificar que algumas anomalias foram corrigidas, outras reduzidas e outras ainda preservadas. Isso nos levará a ter de buscar mais um nível de normalização.

- INSERÇÃO:

Só é possível incluir um novo fornecedor a partir de um pedido.

A análise dessa anomalia já foi feita na primeira forma normal. Temos aqui, ainda presente, a mesma anomalia.

Se incluirmos um pedido novo com CGC DO FORNECEDOR = 8242888-80 e com o NOME DO FORNECEDOR = "Casa dos Softwares", qual será, no futuro, a denominação válida para esse fornecedor? "Casa Software" ou "Casa dos Softwares"?

A análise dessa anomalia já foi feita na primeira forma normal. Temos aqui, ainda presente, a mesma anomalia.

- ELIMINAÇÃO:

Se eliminarmos o pedido 005, perderemos toda a informação sobre o fornecedor "Computerland".

A análise dessa anomalia já foi feita na primeira forma normal. Temos aqui, ainda presente, a mesma anomalia.

- ATUALIZAÇÃO:

Se a data do pedido 003 tiver de ser alterada de 20-Jan. para 21-Jan., teremos que efetuar essa operação sobre várias linhas da tabela.

Essa anomalia aparecia na primeira forma normal e agora, após o processo de normalização da segunda forma normal, foi corrigida, não mais existindo.

Se algum dado do fornecedor "Brasilsoftware" mudar teremos que atualizar diversas linhas da tabela, uma para cada pedido onde ele apareça.

Anteriormente, a extensão das atualizações era muito maior do que agora. Diminuímos consideravelmente o nível de redundância na tabela, mas ainda temos de atualizar diversas linhas da tabela se o dado de um fornecedor for alterado. Portanto, a anomalia foi reduzida mas ainda existe.

Terceira Forma Normal (3FN)

Visando a transformar a estrutura de dados obtida até a segunda forma normal em uma estrutura isenta das anomalias constatadas, deveremos buscar a terceira forma normal. Como já dissemos na primeira forma normal, nem toda tabela precisará ser transformada pela aplicação dos passos da 2FN e 3FN. Algumas tabelas poderão já, após a obtenção da 2FN, estar automaticamente na terceira forma normal. Para verificar se ela já se encontra na terceira forma normal deveremos analisar a tabela com base na definição a seguir

Definição da Terceira Forma Normal

Uma tabela está na terceira forma normal se está na 2FN, e se nenhuma coluna não pertencente à chave fica determinada transitivamente por esta.

"Se está na segunda forma normal ..."

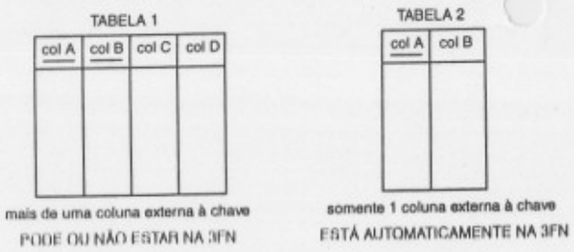
Também nesse ponto é importante garantir que os passos anteriores (1FN e 2FN) já foram realizados devidamente. Sem executá-los previamente não haverá sentido em aplicar a 3FN.

"... se nenhuma coluna não pertencente à chave..."

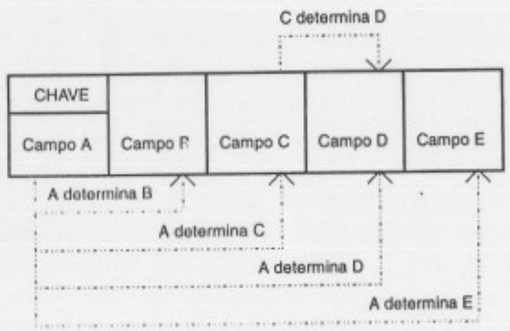
Esse enunciado deixa claro que nessa atividade só deverão ser analisadas as colunas não pertencentes à chave. Independentemente de se nossa chave é composta por uma ou diversas colunas deveremos omitir essas colunas da análise. As colunas que nos interessam para a análise são somente as não formadoras da chave primária da tabela.

"... fica determinada transitivamente por esta."

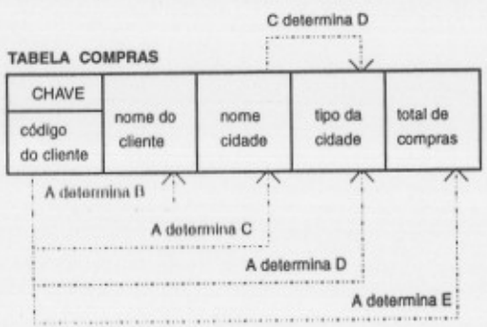
A dependência transitiva de uma chave só será possível se a tabela tiver ao menos duas colunas não pertencentes à chave. Caso tenhamos somente uma coluna externa à chave, essa tabela já estará automaticamente na terceira forma normal.



Mas o que significa exatamente uma dependência transitiva da chave? Dizemos que uma coluna depende transitivamente da chave se seu valor é determinado pelo conteúdo de uma coluna não chave que, por sua vez, também já é determinada pela chave primária da tabela. Em outras palavras temos que: C é determinado pelo valor de B que, por sua vez, já teve seu valor determinado pelo valor de A. Graficamente podemos representar essa situação do seguinte modo:



Nesse exemplo, perceba que o valor da coluna D é determinado pela coluna C, que por sua vez teve seu valor determinado pela coluna A. Poderíamos ter um exemplo concreto que representasse essa situação do seguinte modo:



Nesse exemp' emos uma tabela de COMPRAS dos clientes de uma grande loja de departamentos. Todo cliente é avaliado em função do seu montante anual de compras. Essa avaliação leva em consideração a cidade onde ele reside e o tipo da cidade (interior, capital, metrópole, etc.). Essa tabela já se encontra na primeira e na segunda forma normal, porém ainda não está na terceira forma normal. Temos uma dependência transitiva para o campo TIPO DE CIDADE. Ele é determinado através do NOME DA CIDADE, que por sua vez é determinado pelo CÓDIGO DO CLIENTE. Isso é fácil de ser verificado, pois dado um código de um cliente, encontraremos somente um nome de cidade associado a ele (a cidade onde tem residência fixa) e para essa cidade teremos somente um tipo associado a ela (ou ela é interior, ou capital ou metrópole).

Processo para obtenção da Terceira Forma Normal (3FN)

1. Identificar as colunas que não participam da chave primária da tabela.
2. Para cada uma das colunas identificadas, analisar se seu valor é determinado por alguma outra coluna não pertencente à chave.
3. Para as colunas dependentes transitivamente da chave:
   - Criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) que determinou(aram) o valor da coluna analisada. Agregar a essas tabelas as colunas dependentes transitivamente.
   - Excluir da tabela de origem as colunas dependentes transitivamente das chaves mantendo, porém, a coluna determinante da transitividade na tabela.

1. Identificar as colunas que não participam da chave primária da tabela.

O único cuidado a ser tomado nesse passo é identificar a notação utilizada para representar a tabela e sua chave. Nem sempre as colunas-chave estarão agrupadas no início da tabela ou se estiverem nem sempre utilizarão uma notação conhecida. A partir da identificação das colunas que formam a chave, bastará ignorá-las durante o processo de análise e o que restará serão as colunas que nos interessam.

2. Para cada uma das colunas identificadas, analisar se seu valor é determinado por alguma outra coluna não pertencente à chave.

A identificação da dependência transitiva deverá ser feita coluna a coluna. Essa análise poderá ser acelerada com a experiência de cada um no

domínio em que estiver sendo modelado. Para alguns será mais fácil identificar se uma coluna é dependente transitivamente de outra ou não. Já para outros, talvez seja necessário algum apoio externo para essa identificação.

Para se identificar a dependência transitiva de uma coluna, deve-se indagar:

- Qual outra coluna não pertencente à chave poderia determinar o valor dessa coluna?

ou

- O valor dessa coluna pode ser obtido a partir do conhecimento de algum outro valor existente nessa tabela?

Se com alguma dessas perguntas for possível identificar uma ou mais colunas não pertencentes à chave que determinem valores para outras colunas, então teremos caracterizado uma tabela que ainda não está na terceira forma normal.

Uma boa estratégia é utilizar-se de exemplos com valores reais. Pense em valores instanciados nas colunas e imagine como eles estão interligados. Como uns determinam os outros.

3. Para as colunas dependentes transitivamente da chave, criar novas tabelas onde a chave primária será(ão) a(s) coluna(s) que determinou(aram) o valor da coluna analisada, agregando a essas tabelas as colunas dependentes transitivamente, e excluir das tabelas de origem as colunas dependentes transitivamente das chaves, mantendo porém, a coluna determinante da transitividade na tabela.

Essa é a fase ativa desse processo de normalização. Até agora somente tentávamos identificar se a tabela estava ou não na terceira forma normal. Agora estaremos agindo sobre as tabelas que já estão na segunda forma normal, desmembrando-as em novas tabelas. Perceba que não mais agiremos somente sobre a tabela original (aquela da qual partimos). Agora deveremos aplicar essa regra sobre cada uma das tabelas existentes e que já estejam na segunda forma normal, inclusive aquelas obtidas pelo desmembramento da tabela original.

O processo de normalização da terceira forma normal também dará origem a novas tabelas. Os objetos não observados distintamente e colocados juntos em uma mesma tabela serão explicitados em mais um nível de detalhe. Esse processo "garante" uma maior fidelidade ao processo de normalização, complementando o que já foi feito pela segunda forma normal.

Aplicando-se essa regra sobre a estrutura da tabela COMPRAS, vista no exemplo das dependências transitivas, passaríamos a ter as seguintes tabelas após a normalização na terceira forma normal:

TABELA "COMPRAS"

CHAVE			
código do cliente	nome do cliente	nome cidade	total de compras

TABELA "CIDADES"

CHAVE	
nome cidade	tipo da cidade

Perceba que passamos a ter duas tabelas distintas em vez de somente uma. Surgiu pelo processo de normalização a tabela CIDADES. Como o TIPO DA CIDADE é um atributo da cidade, deveríamos ter uma entidade CIDADE em nosso modelo. Agora ela surge naturalmente através do processo de normalização. Como dissemos, essa é uma das finalidades básicas da normalização.

Vamos dar continuidade ao processo de normalização de nosso exemplo onde apresentávamos uma tabela de PEDIDOS não normalizada. Nos passos anteriores já transformamos a estrutura inicial dessa tabela em uma estrutura normalizada na primeira e segunda formas normais. O resultado que obtivemos até o último ponto foi o seguinte:

TABELA 1 — ORIGINAL

Núm. Ped.	Cód. Prod.	Qtde.	Preço 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

TABELA 2 — OBTIDA DURANTE A 2FN

Núm. Ped.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.
003	20-Jan	Casa Software	8242888-80	Lapa	777
004	27-Jan	Brasilsoftware	8047999-70	Itu	49
005	27-Jan	Computerland	7932253-23	Feijó	122
006	14-Mar	Brasilsoftware	8047999-70	Itu	49

TABELA 3 — OBTIDA DURANTE A 2FN

Cód. Prod.	Nome Prod.
033A	DOS
002M	COREL
145J	ABC
083P	ZAPT
029K	WIN

Aplicando-se sobre essas tabelas a regra da terceira forma normal, deveremos:

"Analisar as colunas não chave..."

Essas colunas são:

Na TABELA 1: QUANTIDADE e PREÇO UNITÁRIO.
Na TABELA 2: DATA EMISSÃO, NOME FORNECEDOR, CGC, RUA e NUM-IMÓVEL
Na TABELA 3: nenhuma, pois ela já está na terceira forma normal (só tem uma coluna não pertencente à chave)

"identificando as dependências transitivas..."

Na TABELA 1:

TABELA 1 — ORIGINAL

Núm. Ped.	Cód. Prod.	Qtde.	Preço 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

O valor da coluna PREÇO UNITÁRIO não pode ser definido com base no valor da coluna QUANTIDADE pois para valores de quantidade iguais poderemos ter diferentes preços. Imagine que alguém pode pedir 100 unidades de um produto com preço unitário igual a R$ 5,00 e também outras 100 unidades de outro produto com preço igual a R$ 7,00.

O mesmo é válido em relação à coluna QUANTIDADE poder ser definida pela coluna PREÇO UNITÁRIO. Alguém pode pedir um produto que custa R$ 9,00 e requisitar 40 unidades e ao mesmo tempo pedir outro produto que também custa R$ 9,00, só que requisitando 30 unidades.

Isso demonstra que o valor de uma coluna não pode ser determinado pelo valor da outra, logo não há dependência transitiva para nenhuma delas estando, portanto, a TABELA 1 já na terceira forma normal.

Na TABELA 2:

Dado o valor de uma DATA EMISSÃO não é possível determinar o NOME DO FORNECEDOR, o CGC, a RUA ou o NÚMERO DO IMÓVEL. (Compare os pedidos 004 e 005.)

Núm. Ped.	Data Emiss.	Nome Fornecedor	CGC	Rua	Núm. Imóv.
003	20-Jan	Casa Software	8242888-80	Lapa	777
004	27-Jan	Brasilsoftware	8047999-70	Itu	49
005	27-Jan	Computerland	7932253-23	Feijó	122
006	14-Mar	Brasilsoftware	8047999-70	Itu	49

Também dadas quaisquer outras colunas (NOME FORNECEDOR, CGC, RUA ou NÚMERO DO IMÓVEL) não é possível determinar o valor da coluna DATA EMISSÃO. Logo, a coluna DATA EMISSÃO não está envolvida em qualquer dependência transitiva.

Entretanto, dado o valor da coluna CGC é possível determinar um e só um valor para as colunas NOME FORNECEDOR, RUA e NÚMERO DO IMÓVEL. Isso denota uma dependência transitiva dessas três colunas em relação à coluna CGC.

"...criando novas tabelas onde a chave primária será(ão) a(s) coluna(s) que determinou(aram) o valor da coluna analisada, agregando a essas tabelas as colunas dependentes transitivamente, e excluir das tabelas de origem as colunas dependentes transitivamente mantendo a coluna que determinou a transitividade."

Aplicando essa regra, obteremos as seguintes tabelas:

TABELA 1 — JÁ NA 3FN

Núm. Ped.	Cód. Prod.	Qtde.	Preço 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

TABELA 2 — APÓS A 3FN
(mantendo a coluna CGC que determinou a transitividade e sem as demais colunas que eram determinadas transitivamente)

Núm. Ped.	Data Emiss.	CGC
003	20-Jan	8242888-80
004	27-Jan	8047999-70
005	27-Jan	7932253-23
006	14-Mar	8047999-70

TABELA 3 — JÁ NA 3FN

Cód. Prod.	Nome Prod.
033A	DOS
002M	COREL
145J	ABC
083P	ZAPT
029K	WIN

TABELA 4 - OBTIDA A PARTIR DA TABELA 2
(por dependência transitiva do CGC)

CGC	Nome Fornecedor	Rua	Núm. Imóv.
8242888-80	Casa Software	Lapa	777
8047999-70	Brasilsoftware	Itu	49
7932253-23	Computerland	Feijó	122

Anomalias Resolvidas pela Terceira Forma Normal (3FN)

- INSERÇÃO:

Não há mais necessidade de um novo pedido para incluir um novo fornecedor.

Sempre que se queira incluir um novo fornecedor basta atualizar a TABELA 4 sem que se tenha qualquer envolvimento com a tabela que contém os dados de pedidos. Isso nos dá a correta visão de que existe um "cadastro" de fornecedores e que os pedidos simplesmente referenciam esses fornecedores quando necessário.

Se tentarmos incluir um novo pedido com CGC DO FORNECEDOR = 8242888-80 e NOME DO FORNECEDOR = "Casa dos Softwares", poderemos detectar a inconsistência através da TABELA 4.

Anteriormente poderíamos cadastrar dois pedidos para o mesmo fornecedor e possuir dados diferentes em cada um deles. Agora, como os dados do fornecedor aparecem uma única vez, além de eliminar a redundância existente, teremos meios de verificar a consistência dos dados.

- ELIMINAÇÃO:

Se eliminarmos qualquer um dos pedidos existentes, não mais perderemos toda a informação sobre um determinado fornecedor.

Como os dados do fornecedor não estão mais associados aos pedidos, e sim em uma tabela distinta, não há perda de informações sobre um determinado fornecedor mesmo que todos os pedidos feitos a ele sejam eliminados.

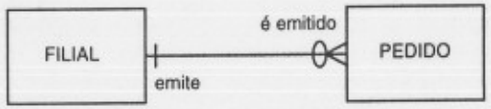
- ATUALIZAÇÃO:

Se algum dado de um fornecedor mudar, teremos que atualizar apenas uma linha da TABELA 4 e todos os pedidos refletirão essa mudança.

Anteriormente deveríamos executar atualizações em cada um dos pedidos existentes para um fornecedor sempre que qualquer um dos dados do fornecedor envolvido mudasse. Além da redundância, corríamos o risco de atualizar alguns pedidos e outros não e então passar a ter inconsistências nos dados disponíveis.

Modelo obtido após a Terceira Forma Normal

Quando iniciamos nosso processo de normalização, partimos de um modelo básico que era:



Agora, após termos aplicado as regras das três formas normais sobre a entidade PEDIDO, obtivemos as seguintes tabelas e atributos:

Tabela	Chave Primária (PK)	Colunas	Entidade
1	NÚM-PEDIDO + CÓD-PRODUTO	QUANTIDADE PREÇO UNITÁRIO	PRODUTO ENCOMENDADO
2	NÚM-PEDIDO	DATA EMISSÃO CGC	PEDIDO
3	CÓD-PRODUTO	NOME PRODUTO	PRODUTO
4	CGC	NOME FORNECEDOR RUA NÚM-IMÓVEL	FORNECEDOR

Observando as tabelas obtidas, podemos constatar que nosso modelo também normalizou-se transformando-se em:

Nesse modelo temos uma representação lógica obtida pela normalização. Mesmo que em nosso modelo conceitual não tivéssemos percebido, inicialmente, todas as entidades necessárias, teríamos tido, através desse processo, o subsídio necessário para identificar as entidades PRODUTO e FORNECEDOR. Com isso poderíamos, inclusive, rever nosso modelo conceitual e atualizá-lo para:

Perceba que a TABELA PRODUTO REQUISITADO, que aparece no modelo lógico, é, na verdade, apenas um relacionamento MxN entre PRODUTO e PEDIDO. Representá-la no modelo conceitual como uma entidade seria introduzir "elementos de implementação" em nosso modelo. Só devemos vê-la como tabela por uma necessidade de implementação.

Por isso preste atenção:

> *Nem todas as tabelas que tenham surgido do processo de normalização deverão ser agregadas a nosso modelo conceitual!*

Identificar quais tabelas realmente representam novas entidades irá envolver a percepção de quais elementos do ambiente observado elas representam. Algumas poderão enquadrar-se efetivamente como entidades, outras como relacionamentos (como no exemplo anterior) e outras até como atributos repetitivos.

Com isso finalizamos a abordagem do processo de normalização. Como já dissemos, existem ainda outras duas formas normais que são citadas na literatura. Recomendamos que aqueles que tenham interesse em conhecê-las consultem a bibliografia no final do livro. O conhecimento das três formas normais, vistas até aqui, com certeza, será o suficiente para que seus projetos atinjam o grau de normalização necessário à implementação sem anomalias.

Derivação de Relacionamentos

A partir desse instante, vamos abordar outro aspecto relativo à derivação de modelos lógicos. Enquanto tratávamos o assunto de normalização, estávamos enfocando as entidades e atributos existentes em nossos modelos conceituais. O tratamento dos relacionamentos, estruturas de Generalização-Especialização e agregação também deverá ser feito durante o processo de derivação. Estes serão nossos próximos pontos de análise.

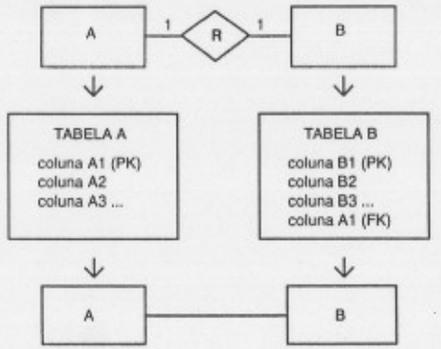
No tocante à derivação de relacionamentos, abordaremos todos os possíveis casos a serem identificados em um modelo conceitual. Veremos os relacionamentos de 1:1, 1:N, M:N, com e sem atributos. As cardinalidades apresentadas serão relativas a valores máximos. Não diferenciaremos o caso de relacionamentos condicionais em cada um dos casos, pois isso não mudará o processo de derivação. Abordaremos ao final, de modo genérico, quais são os impactos da existência de relacionamentos condicionais e como eles deverão ser tratados. Caso você tenha alguma dúvida sobre o que representa cada um desses casos, volte a consultar as páginas que abordaram esses assuntos na modelagem conceitual.

Relacionamentos 1:1 sem Atributos

O caso mais elementar de relacionamento entre duas entidades é o de cardinalidade 1:1. Dentro desse tipo de relacionamento, por sua vez, o caso mais freqüente de ser encontrado é o de relacionamentos 1:1 sem atributos.

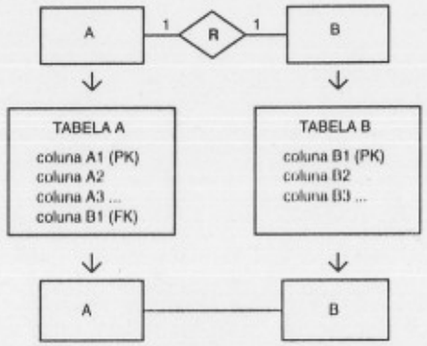
Dentre as quatro alternativas possíveis para derivação desse tipo de relacionamento temos:

1. Acrescer a chave da TABELA A como chave estrangeira na TABELA B.



Para que haja o estabelecimento de um relacionamento das TABELAS A e B, basta que a chave da TABELA A exista na TABELA B. Assim, qualquer elemento de B quando necessitar identificar a qual elemento de A está ligado obterá essa informação através da chave estrangeira da TABELA A que existe na TABELA B.

2. Acrescer a chave da TABELA B como chave estrangeira na TABELA A.

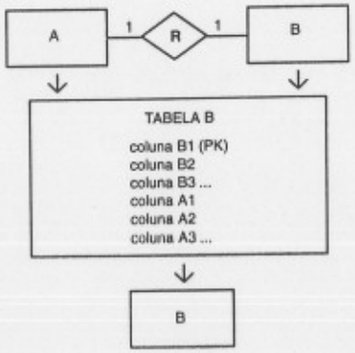


Do mesmo modo que conseguíamos estabelecer um relacionamento entre a TABELA A e B, colocando uma chave estrangeira de A em B, conseguiremos se colocarmos uma chave estrangeira de B em A.

A decisão sobre qual deve ser a chave estrangeira a ser movida para a outra tabela dependerá, em princípio, de qual tabela "nascerá" antes. Se primeiro surgem os dados de A e após isso os de B, então deveríamos migrar a chave estrangeira de A para B. Assim, quando fôssemos criar uma ocorrência na TABELA B já teríamos o valor de A a referenciar na chave estrangeira. O contrário também seria válido se B surgisse antes do que A.

Outro critério que pode ser válido quando as duas entidades têm vida independente, surgindo desvinculadas uma da outra de modo assíncrono, é avaliar o tamanho (em bytes) da chave a ser exportada. Assim, poderíamos exportar a menor das chaves, contribuindo para a redução do espaço físico para armazenamento. Entretanto, isso não deve ser uma regra, somente um critério de desempate quando todas as outras análises de critérios já tiverem sido esgotadas.

3. Migrar os atributos e relacionamentos da TABELA A para a TABELA B excluindo a TABELA A definitivamente.



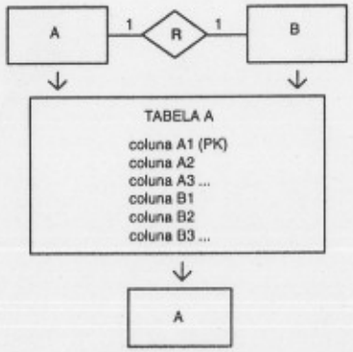
A decisão em utilizar essa estratégia de migração dos dados de uma tabela para outra deverá ter como premissa que os dados de ambas as tabelas, realmente, não têm existência individual. A existência de dados em uma tabela deveria ter sempre a correlação de existência na outra.

Devemos considerar também o fato de que todos os relacionamentos existentes com a ENTIDADE A no modelo conceitual deverão também ser migrados para a ENTIDADE B. Após esse passo então, a ENTIDADE B será transformada em TABELA B, com os atributos de B e A. Na seqüência, os relacionamentos de B (originais e herdados de A) deverão ser derivados aplicando-se as regras para cada um dos graus existentes.

A utilização dessa estratégia, apesar de parecer bastante simples, pode nos levar a problemas futuros. Poderemos detectar, mais a frente, que em alguns casos (ainda não conhecidos durante a fase de projeto) será possível a autonomia entre as tabelas (os dados de uma existirem sem que os da outra existam). Isto nos levará a ter de rever o agrupamento das tabelas e, eventualmente, voltar a separá-las. Poderemos também detectar que o tratamento de alguns relacionamentos que pertenciam a A, agora estabelecidos com B, poderá tornar o entendimento e a manipulação muito mais complexos do que se as tabelas A e B continuassem a existir independentemente.

Por isso, acreditamos que a simples tendência de reduzir o número de tabelas existentes em nosso modelo não deveria ser um bom motivo para agrupar tabelas que possuam relacionamento de 1:1. Talvez essa "economia" venha a "custar caro" no futuro. Se já no modelo conceitual fomos capazes de identificar elementos distintos que se relacionam com grau 1:1, talvez a melhor opção, então, seja mantê-los assim no modelo lógico.

4. Migrar os atributos e relacionamentos da TABELA B para a TABELA A excluindo a TABELA B definitivamente.



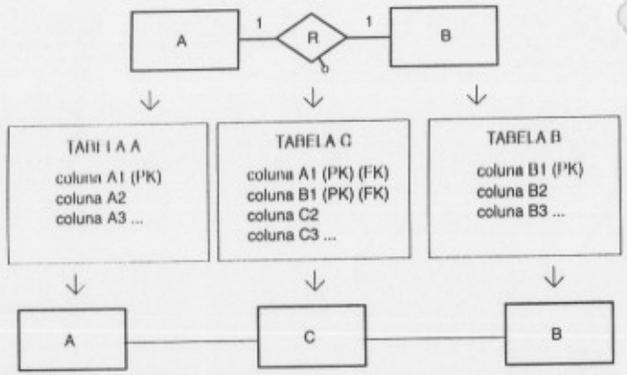
Essa estratégia merece as mesmas considerações da estratégia 3 (vista anteriormente). O único motivo que poderia nos levar a migrar os dados e relacionamentos de B para A, em vez do contrário, seria o menor número de atributos e relacionamentos a serem migrados.

Também como no caso anterior, não consideramos essa estratégia como um grande benefício para o projeto de uma estrutura estável e flexível.

Relacionamentos 1:1 com Atributos

A existência de relacionamentos 1:1 com atributos não é algo muito comum. Poderíamos até dizer que é, realmente, um rigor extremo do projeto conceitual. Entretanto, como no nível conceitual eles poderão existir, apresentaremos três métodos para derivá-los.

1. Criar uma TABELA C para conter as associações entre A e B, agregando as chaves estrangeiras de A e B para dar origem à chave de C, e alocando os atributos do relacionamento na TABELA C.

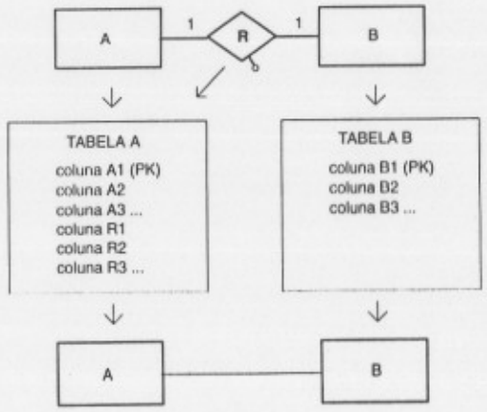


Nesse caso, sempre que a associação entre um elemento da TABELA A for feita com um elemento da TABELA B, criaremos uma linha na TABELA C, estabelecendo essa ligação e armazenando os dados relativos aos atributos do relacionamento.

Devem-se criar meios (triggers, programação ou índices) para que se tenha a garantia de que, desde que um elemento qualquer do conjunto A estabeleça uma primeira ligação com um elemento do conjunto B, esse elemento A não possa mais estabelecer qualquer outra ligação com outro elemento de B, e vice-versa.

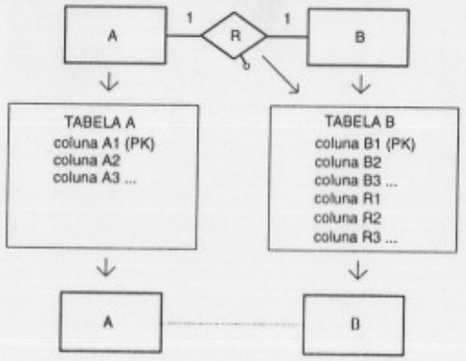
Isso é necessário, pois como a chave primária da TABELA C é a concatenação da chave de A e B, pares como (a1,b1), (a1,b2) e (a2,b1) seriam válidos (não haveria duplicação de chave). Entretanto podemos perceber que através dessas chaves estamos estabelecendo uma relação de M:N entre A e B, pois o elemento a1 se relaciona com dois elementos de B (b1 e b2) e o elemento b1 se relaciona com dois elementos de A (a1 e a2).

2. Migrar os atributos do relacionamento para a TABELA A.



É bastante comum [illegible]car-se essa técnica de derivação quando se trata um relacionamento de 1:1. A migração dos atributos do relacionamento R para uma das tabelas que compõem o relacionamento tem sido quase que uma regra quando se trata esse tipo de caso. O único inconveniente, do ponto de vista teórico, é que os atributos do relacionamento (colunas R1, R2, R3 ...) existirão para todas as linhas da TABELA A, mesmo para aquelas que, efetivamente, não se relacionam com qualquer elemento B. Para que isso seja possível, será necessário definir que essas colunas aceitam valores NULOS. Aí então voltaremos à discussão da aplicabilidade e validade dos valores NULOS em colunas de tabelas. Entretanto, do ponto de vista prático, não há grandes restrições sobre a adoção dessa estratégia.

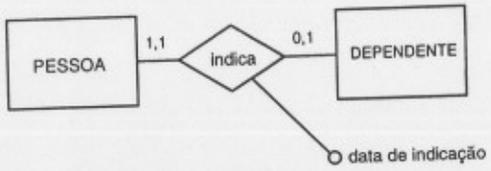
3. Migrar os atributos do relacionamento para a TABELA B.



As mesmas considerações feitas no caso anterior são válidas a essa técnica. A única vantagem em adotar a migração dos atributos do relacionamento R para a TABELA A ou B residirá no fato de que poderemos alocá-los na tabela que tenha suas linhas geradas durante o estabelecimento lógico da relação. Assim, se tivermos um caso onde a TABELA A tem suas linhas já previamente criadas e para estabelecer um relacionamento com a TABELA B é criada a linha em B, então seria apropriado migrar os atributos do relacionamento para B e não para A. Desse modo, evitaríamos manter valores nulos nas colunas da tabela que recebeu os atributos de R.

Exemplo:

Suponha que temos um modelo no qual uma PESSOA pode indicar um, e só um, DEPENDENTE para receber benefícios de um seguro de vida em seu nome (um beneficiário). Nesse caso, deveríamos migrar o atributo DATA DE INDICAÇÃO para a entidade DEPENDENTE, ao invés de migrá-los para a entidade PESSOA.



Perceba que o atributo citado só passa a existir no instante em que um DEPENDENTE é cadastrado e associado a uma PESSOA (que já deve existir previamente). Note também que nem todas as PESSOAS precisam indicar um beneficiário, logo algumas não estabelecerão relacionamento com a entidade DEPENDENTE. Colocar esse atributo na TABELA DEPENDENTE será uma solução ideal que, inclusive, não demandará nem o uso de valores NULOS nas colunas oriundas do relacionamento.

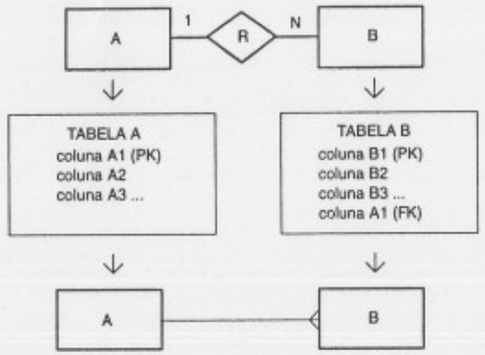
Relações 1:N sem Atributos

Os relacionamentos 1:N representam a grande maioria dos relacionamentos existentes nos modelos. Vêm seguidos em segundo lugar dos relacionamentos M:N e, por último, pelos relacionamentos 1:1.

Também em sua grande maioria, os relacionamentos 1:N não possuem atributos, ou, pelo menos, não são vistos como tendo. Já citamos que a observação e alocação de atributos nos relacionamentos é um formalismo que, apesar de representar exatamente a verdade, somente poucas pessoas adotam. Por isso, consideramos que a derivação de relacionamentos 1:N sem atributos, que será vista neste item, representará a grande parte dos casos de derivação que encontraremos. Felizmente, a regra de derivação desse caso é única e bastante simples, não deixando margens a interpretações ou aberturas a escolhas.

A regra diz que, para derivar um relacionamento 1:N sem atributos, devemos:

1. Acrescer a chave primária da TABELA A como chave estrangeira na TABELA B.



A simples inclusão da chave estrangeira da TABELA A na TABELA B faz com que o relacionamento de 1:N fique implementado satisfatoriamente. A chave estrangeira existente na TABELA B poderá ou não admitir valores NULOS em função da definição dos graus mínimos dos relacionamentos.

Se o relacionamento for 1,1:0,N não haverá qualquer valor NULO envolvido, pois simplesmente estaremos estabelecendo um caso onde linhas da TABELA A poderão ter seu valor de chave primária em 0, 1 ou N linhas da TABELA B. Quando esse caso for referente a 0 linha simplesmente não teremos a linha na TABELA B, logo não haverá valores NULOS envolvidos.

Se o relacionamento for 1,1:1,N estaremos considerando que todo o valor de chave primária da TABELA A aparecerá em 1 ou mais linhas da TABELA B. Este também é um caso que não envolve valores NULOS na TABELA B.

Se o relacionamento for 0,1:0,N ou 0,1:1:N, teremos situações envolvidas onde algumas das linhas da TABELA B poderão estar ligadas a 0 linha da TABELA A. Isso significa que linhas da TABELA B terão na chave estrangeira da TABELA A valores NULOS, pois não estão associadas a nenhuma linha da TABELA A. Isso não caracterizará uma anomalia, mas simplesmente uma consideração a ser feita durante o projeto da base de dados. Deveremos ou não, dependendo do caso, definir que a chave estrangeira da TABELA A existente na TABELA B aceitará valores NULOS.

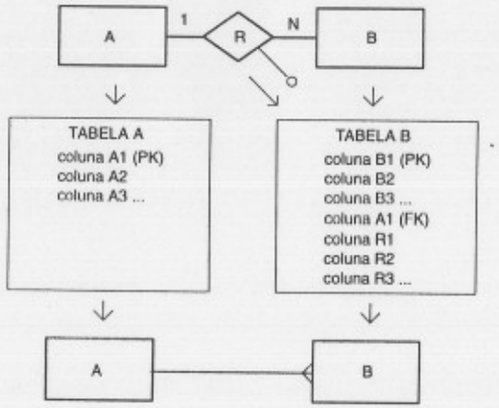
Relacionamentos 1:N com Atributos

Para a derivação desses tipos de relacionamentos, podem ser aplicadas uma das estratégias a seguir:

1. Migrar os atributos do relacionamento para a entidade

Essa primeira alternativa é, na verdade, aquela que, freqüentemente, já é utilizada como simplificação no próprio modelo conceitual. Vimos que muitas pessoas são, naturalmente, induzidas a agregar os atributos do relacionamento 1:N como atributos da entidade B. Caso isso não tenha ocorrido durante o processo de modelagem conceitual, pode-se, nesse instante (na derivação do modelo lógico), optar pela migração. Assim, todos os atributos do relacionamento 1:N passam a fazer parte da TABELA B assumindo valores nulos enquanto o relacionamento entre A e B não se estabelecer efetivamente.

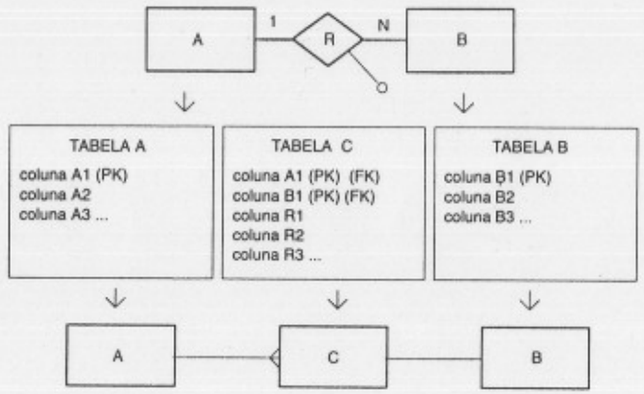
A única desvantagem desse método é a necessidade de definição de valores nulos como aceitáveis para as colunas de R. Persistirá em alguns casos, futuramente, dúvida sobre a interpretação do significado desses valores nulos se alguma referência explícita a essa opção de derivação não for feita no dicionário de dados. Com exceção desse aspecto, não haverá maiores inconvenientes na adoção dessa estratégia de derivação. Hoje, pode-se dizer que esse modo de derivação é um padrão largamente aceito.



2. Criar uma TABELA C para conter as associações entre A e B e alocar os atributos do relacionamento nessa nova tabela.

Outra opção, não tão freqüentemente explorada mas extremamente correta sob o ponto de vista de projeto, é a manutenção da estrutura concebida no modelo conceitual, efetuando-se, então, a criação de uma TABELA C para a implementação dos atributos do relacionamento 1:N.

Nesse caso, a TABELA C manterá como colunas os atributos da relação R e agregará para a formação de sua chave primária as chaves estrangeiras das TABELAS A e B. A incorporação das chaves estrangeiras, na verdade, será o elo de ligação entre as linhas de A e de B (o próprio relacionamento que desejamos implementar). Assim, se desejarmos ligar o elemento A1 com B1, B3 e B5, e o elemento A2 com B2, B4 e B6, bastará implementar linhas na TABELA C com os pares: A1B1, A1B3, A1B5, A2B2, A2B4 e A2B6. Cada uma dessas linhas da tabela C, formada pelas referências a valores de linhas das TABELAS A e B, representará o relacionamento existente entre as linhas das TABELAS. A eles agregaremos os valores dos atributos do relacionamento, instanciando os valores das colunas da TABELA C correspondentes.



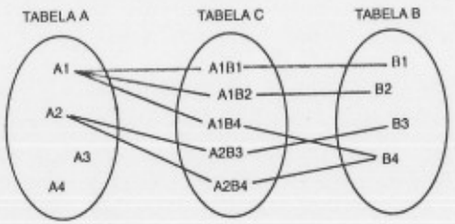
Perceba que após a derivação nosso modelo lógico passa a ter um relacionamento de 1:N entre as tabelas A e C (implementado pela migração da chave primária da TABELA A como chave estrangeira na TABELA C) e um outro relacionamento de 1:1, entre as TABELAS B e C (implementado também pela migração da chave primária da TABELA B como chave estrangeira na TABELA C).

Se analisarmos agora sob outro enfoque o modelo lógico obtido, podemos interpretar da seguinte maneira: Cada ocorrência da TABELA A estabelece N relacionamentos com B através de C, e cada ocorrência de B estabelece um único relacionamento com A através de C. A TABELA C passa a ser somente um elemento de conexão (relacionamento) que é gerado no exato instante em que desejamos relacionar as TABELAS A e B.

O principal ponto a ser observado nessa implementação é que será necessário garantir através do SGBD ou de meios externos (triggers, programação, etc.) que o relacionamento 1:1 entre as TABELAS B e C realmente mantenha esta cardinalidade máxima. Se não tivermos garantias de que, a partir do instante em que uma instância da TABELA B estabeleça

um relacionamento com uma instância da TABELA C, nenhum outro relacionamento entre essa mesma instância de B possa ser estabelecido com qualquer outra instância de C, estaremos transformando nossa relação original 1:N com atributos em uma relação M:N com atributos.

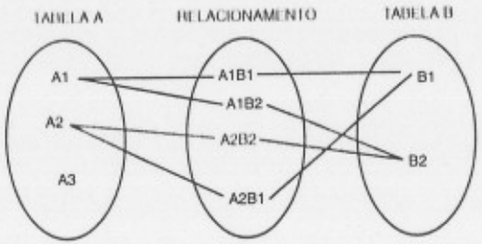
Isso pode ser facilmente observado se analisarmos as seguintes associações:



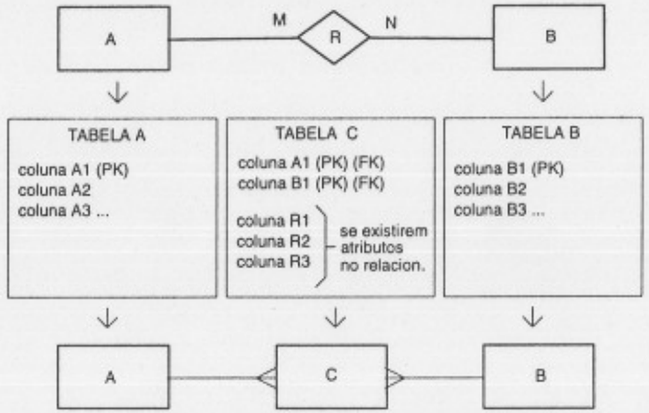
Como não pudemos garantir o relacionamento 1:1 entre as TABELAS B e C (ele na verdade ficou sendo 1:N) também não pudemos garantir que todos os elementos de B tivessem somente uma associação com um elemento da TABELA A (o elemento B4 aparece relacionando-se com A1 e com A2). Para que isso não ocorra será necessário definir na TABELA C, além do índice natural de sua chave primária (formado pelas colunas A1 e B1), um índice sobre a coluna B1, indicando para essa coluna a não permissão de valores duplicados. Perceba que somente utilizando-se o índice natural da chave primária da TABELA C não conseguiríamos restringir a duplicação de B, pois mesmo no exemplo dado anteriormente, apesar da anomalia encontrada, não teríamos duplicado a chave primária da TABELA C.

Relacionamentos M:N com ou sem Atributos

Para a derivação dos relacionamentos M:N, independentemente da existência ou não de atributos nesse relacionamento, deveremos aplicar uma mesma estratégia. Essa estratégia será baseada no fato de que se um relacionamento M:N representa a interseção dos elementos de um conjunto A com um conjunto B, então devemos representar fisicamente os possíveis pares formados pela agregação dos elementos de A e B.



Assim teremos que para derivar um relacionamento M:N entre as TABELAS A e B devemos criar sempre uma TABELA C, agregando as chaves estrangeiras de A e B para formar a chave primária da TABELA C. Caso existam atributos no relacionamento, estes deverão passar a fazer parte da TABELA C como simples colunas (não participantes da chave).



Nessa implementação não haverá necessidade de se garantir qualquer tipo de restrição de armazenamento, além da restrição já imposta pela própria chave primária da TABELA C (não duplicação de chave). Qualquer elemento da TABELA A, por definição, poderá compor "pares" com qualquer elemento da TABELA B e vice-versa. A única impossibilidade, como já mencionamos, é um elemento da TABELA A compor duas vezes o mesmo "par" com o mesmo elemento do conjunto B.

Derivação de Auto-relacionamentos

A derivação de auto-relacionamentos muitas vezes causa certa preocupação e dificuldade de entendimento aos menos experientes. Já percebemos que grande parte dessa dificuldade é justamente advinda do fato de se procurar dar um tratamento "muito diferente" a esses tipos de relacionamentos.

Para tratá-los sem dificuldade, uma estratégia bastante simples pode ser utilizada:

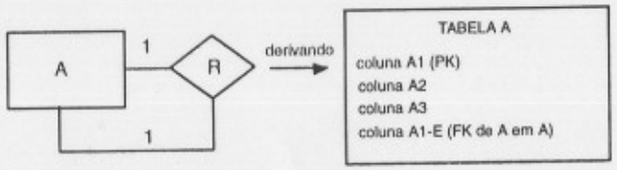
1. Ao derivar um auto-relacionamento, esqueça que se trata de um relacionamento diferente dos demais.
2. Imagine ter um relacionamento entre duas entidades distintas A e A (duas entidades incorporadas em uma mesma).

3. Sempre que as regras de derivação (já vistas anteriormente) struírem "migre a chave de A para B", "incorpore a chave de A em B", "componha a chave de C a partir das chaves de A e B", etc., faça a operação indicada usando no papel de entidade A e B sempre a mesma entidade A.

Para que essa estratégia fique mais clara, daremos alguns exemplos simples de sua utilização:

1. Auto-relacionamento 1:1

Assim como nos demais relacionamentos, a regra diz que se deve: "migrar a chave da TABELA A para a TABELA B". Logo, migraremos a chave da TABELA A para a própria TABELA A.

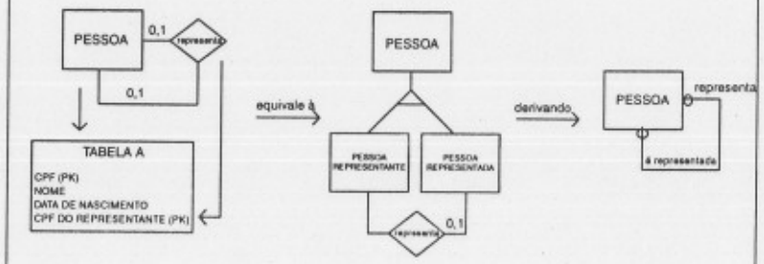


Perceba que se o relacionamento 1:1 fosse entre A e B, teríamos migrado para a TABELA A a chave estrangeira de B. Como a TABELA B, nesse caso, é a própria TABELA A, migramos a chave estrangeira representada pela coluna A1 para a própria TABELA A. Nesse caso, precisaremos passar a diferenciar o nome da coluna criada para que não tenhamos duas colunas A1.

Todas as linhas da TABELA A terão uma nova coluna A1-E, na qual, caso haja o relacionamento de um elemento A com outro elemento A, teremos o valor da chave relacionada.

Exemplo:

Suponha um modelo onde temos um relacionamento de 1:1 entre PESSOA e PESSOA. Esse relacionamento pode significar, por exemplo, "pessoa representa outra pessoa", como no caso de uma procuração. Estamos assumindo, em nosso exemplo, que se uma pessoa é procuradora de uma outra, não será de mais ninguém e que se uma pessoa dá uma procuração a outra pessoa também não dá a mais ninguém. Nesse caso, estamos criando, em nosso modelo, um relacionamento de 1:1 para a entidade PESSOA.



No caso do relacionamento 1:1, vimos que podemos migrar tanto a chave da TABELA A para a TABELA B ou vice-versa. Nesse caso, como só temos uma tabela envolvida, teremos sempre a criação de uma nova coluna igual à chave primária da própria tabela. A única diferença de interpretação que deveremos fazer será definir se a coluna criada representará um ou outro papel do relacionamento. Em nosso exemplo poderíamos ter definido a nova coluna criada como:

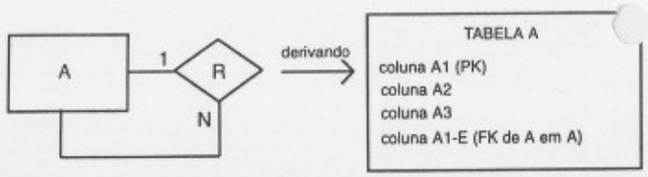
CPF DO REPRESENTANTE (tal como usamos) ou
CPF DO REPRESENTADO

Assim, se Paulo representa Mário, poderíamos ter duas opções válidas de implementação:

- Na linha relativa aos dados de Mário, teríamos o CPF DO REPRESENTANTE, que nesse caso seria o de Paulo.
- Na linha relativa aos dados de Paulo, teríamos o CPF DO REPRESENTADO, que nesse caso seria o de Mário.

2. Auto-relacionamento 1:N

Também nesse caso, tal como nos demais relacionamentos 1:N, deveremos executar a regra que diz que: "migrar a chave da TABELA A para a TABELA B". Logo, migraremos a chave da TABELA A para a TABELA A.

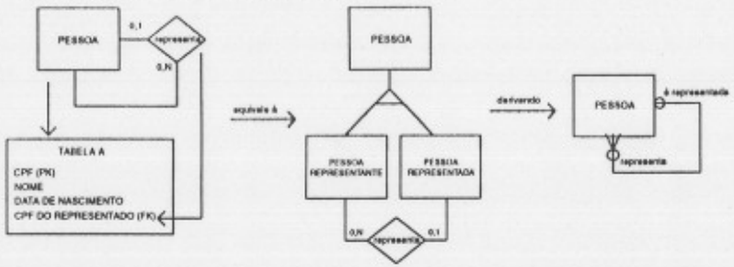


Perceba que se o relacionamento 1:N fosse entre A e B, teríamos migrado para a TABELA A a chave estrangeira de B. Como a TABELA B, nesse caso, é a própria TABELA A, migramos a chave estrangeira representada pela coluna A1 para a própria TABELA A. Nesse caso, também como no caso anterior, precisaremos passar a diferenciar o nome da coluna criada para que não tenhamos duas colunas A1.

O grande diferencial entre essa estratégia e a estratégia do caso 1:1 é que, no caso anterior, poderíamos escolher se a migração de chaves se dava da TABELA A para B ou de B para A, o que fazia com que os papéis do campo CPF REPRESENTANTE OU REPRESENTADO se alterasse. Agora, por se tratar de um relacionamento 1:N, teremos sempre que migrar a chave da TABELA A para a TABELA B, sem escolha.

Exemplo:

Suponha o mesmo modelo tratado anteriormente. Passaremos, entretanto, agora a ter um relacionamento de 1:N entre PESSOA e PESSOA. Esse relacionamento terá o mesmo significado anterior, mas mudaremos a regra de negócio a ele associada. Agora, uma pessoa poderá autorizar várias outras pessoas através de procurações distintas. Porém, caso uma pessoa seja representante de alguém não poderá ser de mais ninguém. Com essa nova regra, passamos a ter um relacionamento de 1:N.



Perceba que se uma pessoa pode ser representada por diversas outras, não poderíamos criar na tabela PESSOA uma coluna chamada CPF DO REPRESENTANTE. Se fizéssemos isso teríamos de ter uma coluna com múltiplos valores. Já se na tabela PESSOA, criarmos uma coluna CPF DO REPRESENTADO, poderemos indicar, sempre que necessário o CPF da única pessoa representada por alguém. Vejamos através de valores explícitos esse caso.

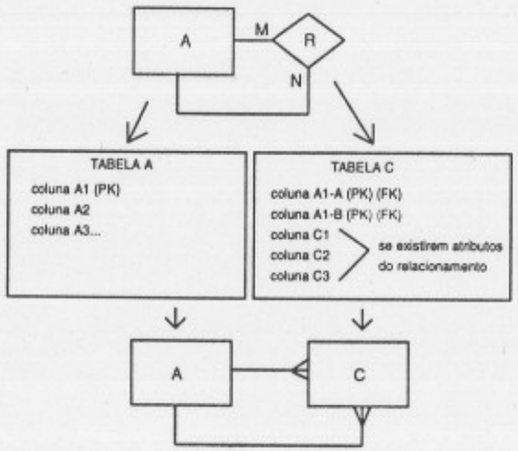
Esse diagrama mostra esquematicamente a tabela PESSOA, suas linhas e seus relacionamentos. Temos estabelecido:

É REPRESENTADO	É REPRESENTANTE
João	José
João	Maria
João	Carlos
Nelson	Eduardo

Logo, como dissemos, na linha da tabela PESSOA relativa a João, não poderíamos ter o CPF DE SEU REPRESENTANTE pois existem três representantes (José, Maria e Carlos). Entretanto, se colocarmos na linha relativa a José uma coluna indicativa do CPF DO REPRESENTADO poderemos indicar através dessa coluna o CPF de João e isto será suficiente pois José não é mais representante de ninguém (por determinação de nosso negócio).

3. Auto-relacionamento M:N

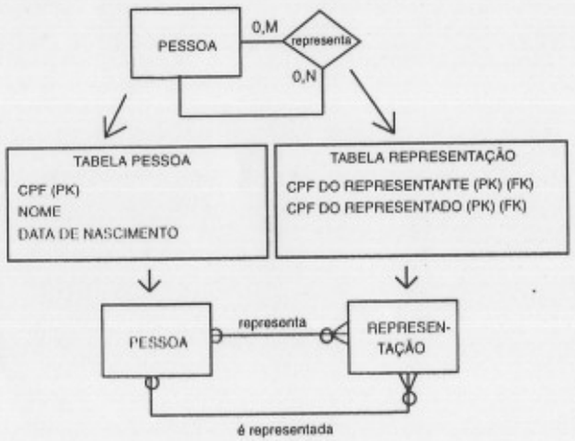
Seguindo a mesma indicação dada anteriormente para derivação de relacionamentos M:N, bastará que criemos uma nova tabela e agreguemos as chaves das TABELAS A e B para dar origem à chave primária de C.



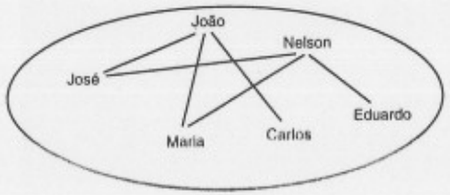
Observe que na TABELA C criamos duas colunas, de nome A[illegible] e A1-B, a partir da migração da chave primária da TABELA A. Cada uma das colunas é originária da TABELA A no papel de entidade A e B respectivamente. Através destas colunas poderemos criar os "pares" indicativos de todas as associações estabelecidas entre um elemento da TABELA A e outro elemento, também da TABELA A.

Exemplo:

Prosseguindo com as transformações em nosso exemplo já abordado anteriormente, vamos admitir agora que uma pessoa possa ser representada por diversas outras e que um representante, por sua vez, possa representar diversas outras pessoas. Com isso teremos um relacionamento M:N. Aplicando a regra conhecida obteremos:



Graficamente, poderíamos ter a seguinte situação:



Esse diagrama mostra esquematicamente a tabela PESSOA, suas linhas e seus relacionamentos. Temos estabelecido:

É REPRESENTADO	É REPRESENTANTE
João	José
João	Maria
João	Carlos
Nelson	José
Nelson	Maria
Nelson	Eduardo

Compare agora os pares mostrados no exemplo com a estrutura da TABELA C. Você perceberá que a existência das colunas CPF DO REPRESENTADO e CPF DO REPRESENTANTE espelham exatamente nossa necessidade de estabelecimento de ligações.

Derivação de Estruturas de Generalização-Especialização

O processo de derivação de uma estrutura de Generalização-Especialização não nos levará diretamente a um modelo lógico para a implementação. Ele, na verdade, nos fornecerá um novo modelo conceitual simplificado onde a estrutura, antes existente, será substituída, simplesmente, por entidades e relacionamentos comuns. Estaremos substituindo a notação especial utilizada para representar essa estrutura por elementos comuns (entidades e relacionamentos) e, após isso, aplicando sobre eles as mesmas regras de derivação já vistas. A partir desse ponto teremos obtido os elementos lógicos resultantes da derivação.

Esse tipo de estratégia é útil para que retiremos a complexidade de derivação das estruturas consideradas "especiais" (estruturas de Generalização-Especialização). Se conseguirmos reduzir essas estruturas às estruturas convencionais (entidades e relacionamentos), poderemos, então, aplicar as regras básicas já conhecidas para obter, finalmente, o modelo lógico.

Basicamente, a decisão a ser tomada no processo de derivação de estruturas de Generalização-Especialização diz respeito ao grau de detalhamento a ser mantido no modelo lógico. Vimos anteriormente que esse tipo de estrutura procura explicitar detalhes pertinentes a subconjuntos de um conjunto maior. Durante o processo de derivação, poderemos adotar quatro diferentes estratégias para tratar essas estruturas:

1. Criar uma tabela para a entidade generalizada e uma tabela para cada entidade especializada.
2. Criar somente uma tabela para a entidade generalizada e migrar todos os atributos e relacionamentos especializados para essa tabela.

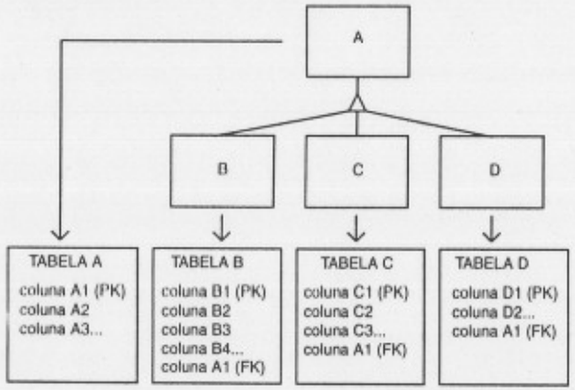
3. Criar somente tabelas para as entidades especializadas e n ar todos os atributos e relacionamentos generalizados para cada uma dessas tabelas.

4. Utilizar as estratégias 3 e 4 de maneira combinada, migrando atributos e relacionamentos das entidades especializadas para a entidade generalizada e/ou vice-versa.

Cada uma dessas opções apresenta vantagens e desvantagens e todas elas resultam em modelos lógicos válidos sob o ponto de vista de fidelidade ao modelo conceitual. A escolha, em cada um dos casos, deverá levar em consideração fatores tais como facilidade de manuseio, facilidade de manutenção de integridade, espaço físico para alocação, grau de redundância desejado, estruturas de dados de cada uma das entidades, etc. Será difícil estabelecer regras rígidas para o processo de escolha, mas tentaremos, durante a explicação de cada um dos casos, dar algumas indicações de possíveis pontos fortes que denotem sua aplicação.

1. Criar uma tabela para a entidade generalizada e uma tabela para cada entidade especializada, mantendo os relacionamentos associados a cada uma delas.

Essa primeira opção define que o modelo conceitual, onde aparecem uma entidade generalizada e várias entidades especializadas, será transportado diretamente para um modelo lógico onde aparecerá uma tabela para cada entidade existente.



Entretanto, para que isso possa ser feito, deveremos ter algum elo de ligação entre a TABELA A (mãe) e as demais TABELAS B, C e D (filhas). Não bastaria criar somente uma tabela para cada entidade existente no

modelo conceitual sem que fosse mantida a semântica de que "a partir de uma ocorrência existente na entidade generalizada podemos obter uma ocorrência na entidade especializada e vice-versa".

Para que seja possível ligar a tabela generalizada com as tabelas especializadas, deveremos criar um relacionamento "artificial" (que somente traduz a especialização) entre elas. Esse relacionamento será de grau 1:1 e deverá ser estabelecido entre a entidade generalizada e cada uma das entidades especializadas.

A partir desse instante, teremos recaído no caso comum de um relacionamento 1:1 entre duas entidades. Poderemos então analisar as possíveis alternativas de derivação desse relacionamento e realizar nossa opção.

Perceba, portanto, que a seqüência básica para derivar uma estrutura de Generalização-Especialização usando essa estratégia é bastante simples:

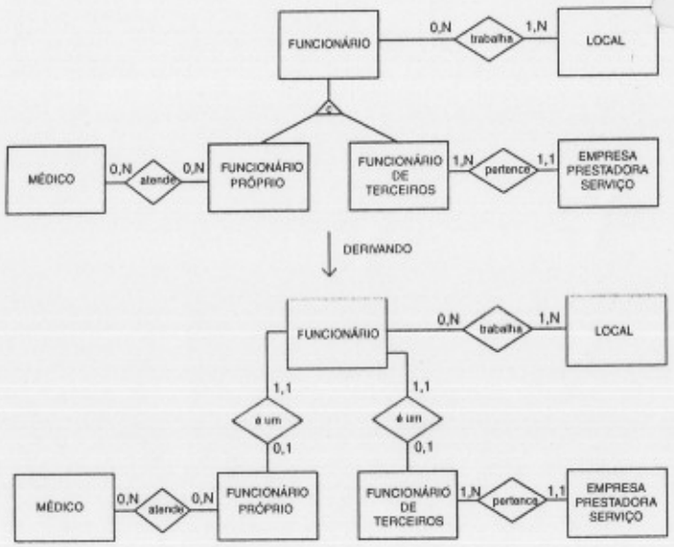
1. Observar o modelo conceitual original.

2. Onde for identificada uma estrutura de Generalização-Especialização substituí-la por um relacionamento de 1:1 entre cada uma das entidades especializadas e a entidade generalizada (somente durante a derivação — Lembre-se de que o modelo conceitual deverá continuar mantendo a estrutura de Generalização-Especialização para manter sua semântica).

3. Passar a tratar cada uma das entidades e relacionamentos obtidos de forma convencional como se não mais tivessem qualquer característica especial.

Exemplo:

Vamos supor um modelo de uma empresa que possui diversos funcionários alocados em seu quadro funcional. Cada um deles pode executar suas atividades em diversos locais distintos dentro da empresa. Entretanto, estes funcionários dividem-se em dois tipos: FUNCIONÁRIO PRÓPRIO e FUNCIONÁRIO DE TERCEIROS. Somente os do tipo FUNCIONÁRIO PRÓPRIO podem utilizar os serviços médicos oferecidos pela empresa. Por outro lado, somente os do tipo FUNCIONÁRIO DE TERCEIROS precisam ser identificados a que EMPRESAS PRESTADORAS DE SERVIÇO pertencem. Também em nível de atributos, temos características comuns (atributos que se aplicam a todos os funcionários) e características distintas (atributos que só se aplicam aos do tipo FUNCIONÁRIO PRÓPRIO ou só aos FUNCIONÁRIOS DE TERCEIROS). Através da análise das características das entidades, dos atributos e dos relacionamentos existentes, optou-se por derivar o modelo mantendo a existência de três tabelas distintas. Assim temos:



Observação:

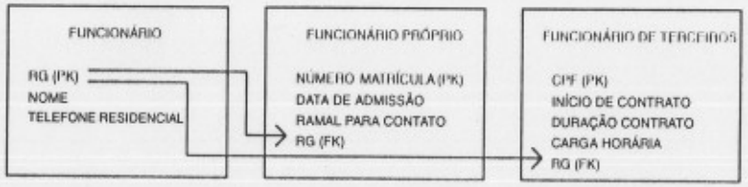
- Todos os atributos comuns (existentes na entidade FUNCIONÁRIO) serão alocados em uma tabela criada para representar essa entidade.
- Os atributos específicos da entidade FUNCIONÁRIO PRÓPRIO serão alocados em uma segunda tabela, que será criada para representar somente essa entidade.
- Os atributos específicos da entidade FUNCIONÁRIO DE TERCEIROS serão alocados em uma terceira tabela também criada para representar somente essa entidade.
- Os relacionamentos existentes em cada uma das entidades serão mantidos associados a cada uma das tabelas que as representam.

Estratégia para o estabelecimento da ligação entre as entidades:

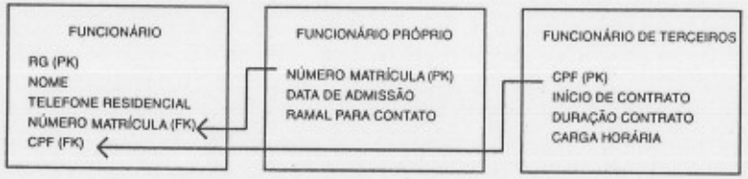
Pode-se recomendar como uma boa estratégia exportar a chave primária da tabela FUNCIONÁRIO para as demais tabelas (durante a derivação do relacionamento 1:1), pois isso garantirá homogeneidade no tratamento independentemente das características das chaves primárias das tabelas FUNCIONÁRIO PRÓPRIO e FUNCIONÁRIO DE TERCEIROS. Assim, se FUNCIONÁRIO PRÓPRIO for identificado por NÚMERO DA MATRÍCULA e FUNCIONÁRIO DE TERCEIROS for identificado por CPF, não haverá necessidade de migrar essas duas chaves para a tabela FUNCIONÁRIO e tornar sempre uma delas nula.

A segunda estratégia possível porém menos recomendada, é migrar para a tabela generalizada (FUNCIONÁRIO) as chaves estrangeiras de cada uma das tabelas especializadas (FUNCIONÁRIO PRÓPRIO e FUNCIONÁRIO DE TERCEIROS). Isso fará com que sempre uma das chaves estrangeiras tenha valor nulo quando o tipo de especialização for "categoria" e não "papel".

ESTRATÉGIA RECOMENDADA:



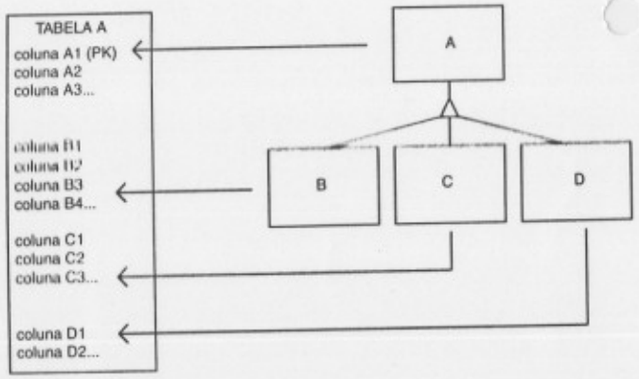
ESTRATÉGIA CORRETA PORÉM POUCO RECOMENDADA:



Perceba que nessas tabelas apenas citamos as colunas e chaves estrangeiras (FK) envolvidas na derivação da estrutura de Generalização-Especialização (relacionamento 1:1). Poderão vir a surgir mais colunas decorrentes de outras derivações de relacionamentos que existam no modelo.

2. Criar somente uma tabela para a entidade generalizada e agregar os atributos e relacionamentos especializados a essa entidade.

Quando observarmos que os atributos e relacionamentos especializados são em pequeno número e que, quase sempre, terão valores associados (não nulos), poderemos optar por não manter uma tabela para cada uma das entidades que compõem a estrutura de Generalização-Especialização. Poderemos, neste caso, migrar os atributos e relacionamentos, existentes no modelo conceitual, que estão associados às entidades especializadas para a entidade generalizada e, depois, derivar a estrutura obtida de modo convencional.

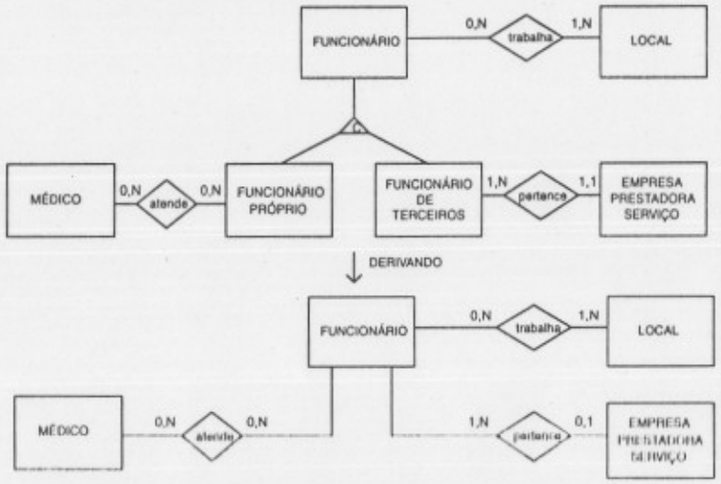


Nesse caso, não teremos necessidade de estabelecer qualquer tipo de vínculo entre as entidades A, B, C e D já que todas deram origem a uma única TABELA A. A presença de colunas na TABELA A oriundas de cada uma das entidades já é o próprio elo de ligação. Entretanto, um ponto importante a ser observado nesse tipo de estratégia de derivação é que todos os relacionamentos existentes nas entidades B, C e D deverão ser migrados também para a entidade A e, somente após isso, derivados. Podemos, portanto, estabelecer os seguintes passos a serem seguidos durante esse processo de derivação:

1. Observar o modelo conceitual original.
2. Onde for identificada uma estrutura de Generalização-Especialização, substituí-la por uma única entidade que representará somente a entidade generalizada.
3. Transferir os atributos das entidades especializadas para a entidade generalizada.
4. Transferir os relacionamentos das entidades especializadas para a entidade generalizada.
5. Passar a tratar a entidade e os relacionamentos obtidos de forma convencional como se não mais tivessem qualquer característica especial. (Lembrar que essa transformação se aplica somente ao modelo lógico. No modelo conceitual devem ser preservadas as características originais.)

Exemplo:

Utilizando o modelo conceitual já visto no tópico anterior, vamos agora observar qual seria o resultado obtido se optássemos pela manutenção somente da entidade generalizada durante o processo de derivação.



Observação:

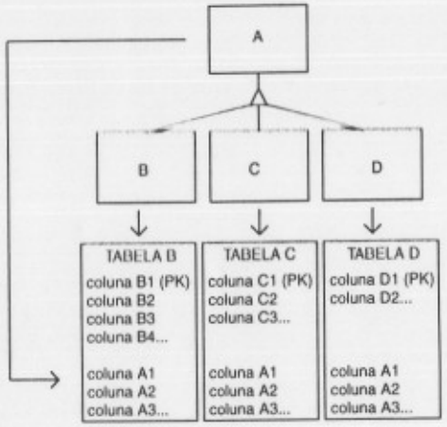
- Todos os atributos comuns (existentes na entidade FUNCIONÁRIO) serão mantidos na tabela FUNCIONÁRIO.
- Os atributos específicos da entidade FUNCIONÁRIO PRÓPRIO serão migrados para a tabela FUNCIONÁRIO (passando a ter valores nulos quando se tratar de um FUNCIONÁRIO DE TERCEIROS).
- Os atributos específicos da entidade FUNCIONÁRIO DE TERCEIROS serão também migrados para a tabela FUNCIONÁRIO (passando a ter valores nulos quando se tratar de um FUNCIONÁRIO PRÓPRIO).
- Os relacionamentos existentes em cada uma das entidades especializadas serão migrados para a entidade generalizada. Perceba que a associação entre EMPRESA PRESTADORA DE SERVIÇO e FUNCIONÁRIO passa a ser não mais de 1,1:1,N mas sim 0,1:1,N pois alguns funcionários (aqueles que forem do tipo próprios) não estarão ligados a nenhuma empresa prestadora de serviços.

3. Criar somente tabelas para as entidades especializadas, agregando os atributos e relacionamentos generalizados a essas entidades.

Uma boa estratégia para se evitar a existência de colunas com valores nulos em grande escala na tabela resultante da derivação de uma estrutura

de Generalização-Especialização é migrar, ao contrário do caso anterior, os atributos generalizados para cada uma das entidades especializadas. Com isso, obteremos uma tabela para cada entidade especializada em que constarão, além de seus atributos e relacionamentos próprios, aqueles incorporados pela migração das características generalizadas para essa tabela.

O fator a ser considerado nessa opção é que deverão ser replicados os atributos e relacionamentos generalizados em tantas entidades especializadas quantas houver. Isso fará com que processos de manutenção desses atributos e relacionamentos também tenham que ser individualizados em cada uma das tabelas resultantes da derivação final.



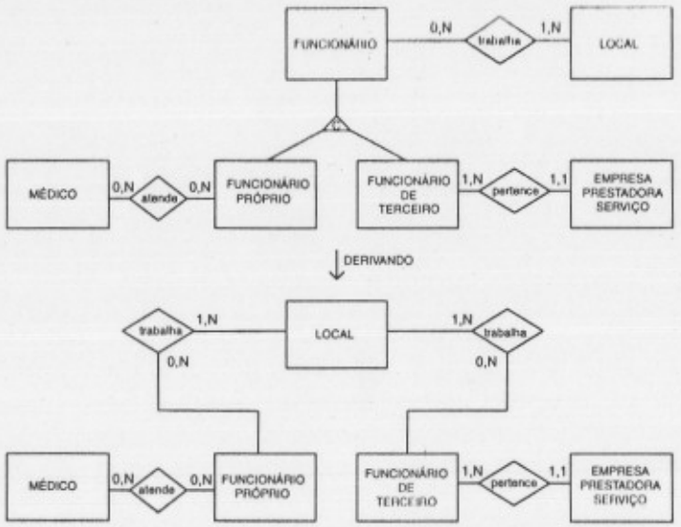
Nesse caso, não teremos, também, necessidade de estabelecer qualquer tipo de vínculo entre as entidades A, B, C e D, já que a entidade A deixará de existir pela incorporação de seus atributos e relacionamentos a cada uma das entidades B, C e D. Entretanto, um ponto importante a ser observado nesse tipo de estratégia de derivação é que todos os relacionamentos existentes na entidade A deverão ser migrados também para as entidades B, C e D e, somente após isso, derivados. Podemos, portanto, estabelecer os seguintes passos a serem seguidos durante esse processo de derivação:

1. Observar o modelo conceitual original.
2. Onde for identificada uma estrutura de Generalização-Especialização substituí-la somente pelas entidades que representam as entidades especializadas.
3. Transferir os atributos da entidade generalizada para cada uma das entidades especializadas.

4. Transferir os relacionamentos da entidade generalizada para cada uma das entidades especializadas.
5. Passar a tratar cada entidade e cada relacionamento obtidos de forma convencional como se não mais tivessem qualquer característica especial. (Lembrar que essa transformação se aplica somente ao modelo lógico. No modelo conceitual devem ser preservadas as características originais.)

Exemplo:

Continuando a aplicar o modelo conceitual já visto nos tópicos anteriores, vamos agora observar qual seria o resultado obtido se optássemos pela manutenção somente das entidades especializadas durante a derivação.



Observação:

- Todos os atributos comuns (existentes na entidade FUNCIONÁRIO) serão migrados para as entidades especializadas (replicando-os).
- Os atributos específicos da entidade FUNCIONÁRIO PRÓPRIO serão mantidos na tabela FUNCIONÁRIO PRÓPRIO.
- Os atributos específicos da entidade FUNCIONÁRIO DE TERCEIROS serão também mantidos na tabela FUNCIONÁRIO DE TERCEIROS.
- Os relacionamentos existentes na entidade generalizada (FUNCIONÁRIO) serão migrados para as entidades especializadas. Perceba que duplicamos o relacionamento "trabalha" mantendo sua cardinalidade.

4. Migrar alguns atributos e relacionamentos das entidades especializadas para a entidade generalizada e/ou vice-versa.

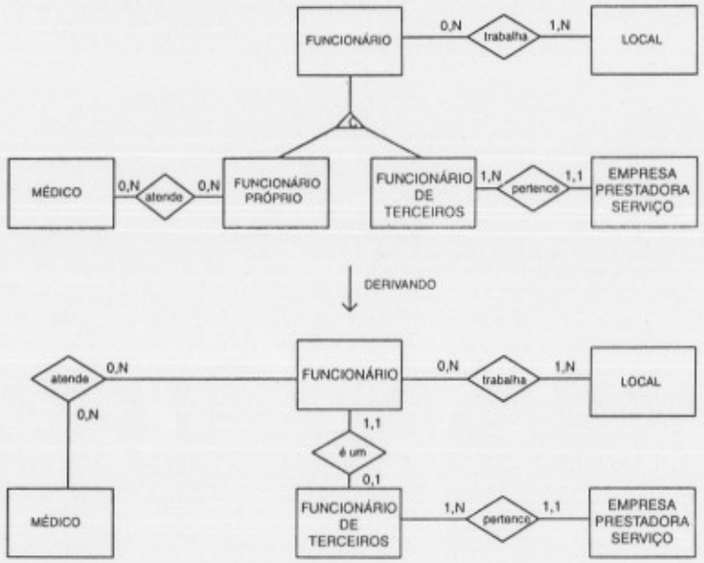
Em alguns casos será pouco interessante migrar todos os atributos e relacionamentos de uma entidade para a outra. Veremos, pela análise caso a caso, que alguns atributos ou relacionamentos poderiam se beneficiar com a migração e outros, não. Deveremos, então, selecionar os candidatos à migração e para eles aplicar os seguintes passos.

1. Manter as entidades especializadas e as entidades generalizadas no modelo, aplicando as regras vistas no item 1 para derivar a estrutura intermediária de derivação (se em cada uma delas houver pelo menos um atributo ou relacionamento).
2. Para os atributos e relacionamentos que tiverem de migrar das entidades especializadas para a entidade generalizada, aplicar as regras vistas no item 2.
3. Para os atributos e relacionamentos que tiverem de migrar da entidade generalizada para as entidades especializadas aplicar as regras vistas no item 3.

Poderão existir casos onde todos os atributos e relacionamentos de uma entidade especializada migrem para a entidade generalizada. Nesse caso, somente a entidade que teve seus atributos e relacionamentos migrados deixará de existir no modelo intermediário da derivação. Em todos os demais casos, sempre que pelo menos um atributo ou relacionamento reste em uma entidade, teremos conservado sua representação no modelo e teremos de ligá-la à entidade generalizada através de um relacionamento 1:1.

Exemplo:

Se no exemplo utilizado até o momento tivéssemos definido que todos os atributos e relacionamentos da entidade FUNCIONÁRIO PRÓPRIO tivessem migrado, mas que houvesse pelo menos um atributo ou um relacionamento na entidade FUNCIONÁRIO DE TERCEIROS, obteríamos o seguinte modelo:



Perceba que, como todos os atributos e relacionamentos exclusivos de FUNCIONÁRIO PRÓPRIO migraram para a entidade FUNCIONÁRIO, deixamos de ter em nosso modelo intermediário de derivação a entidade FUNCIONÁRIO PRÓPRIO. Já a entidade FUNCIONÁRIO DE TERCEIROS, mesmo que tivesse tido alguns de seus atributos migrados para a entidade FUNCIONÁRIO, preservou algumas características ainda em seu domínio, o que fez com que continuasse a ser representada no modelo. Para estabelecer, portanto, um elo de ligação entre ela e a entidade FUNCIONÁRIO, criamos um relacionamento 1:1.

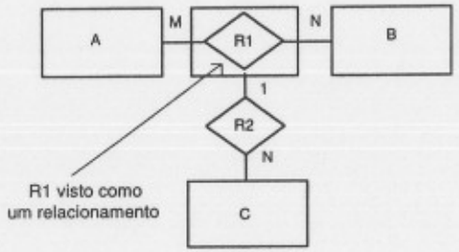
Derivação de Agregações

A agregação foi um elemento introduzido no modelo conceitual para representar de modo mais claro uma relação não-binária. O resultado obtido foi um elemento com características de entidade e de relacionamento simultaneamente. Para derivá-lo iremos tratar primeiramente seu aspecto ligado ao relacionamento e, depois, o aspecto ligado à entidade. (Se você

lembrar, esta também foi a ordem natural em que a agregação foi [illegible]ada durante a modelagem.)

1. Aspecto relativo ao relacionamento

No modelo a seguir temos um caso de agregação entre as entidades A, B e C. Temos estabelecido um relacionamento inicial entre A e B e que posteriormente, visto como entidade, passa a se relacionar com C.

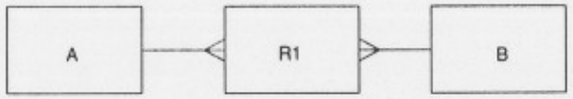


Dizer que iremos tratar primeiramente o aspecto relativo ao relacionamento significa que iremos procurar obter primeiramente um modelo intermediário em que o relacionamento R1 seja resolvido.

Para resolver o relacionamento R1 deveremos observar primeiramente suas características (1:1, 1:N ou M:N, com ou sem atributos). Dependendo da avaliação das características desse relacionamento poderemos optar por uma das estratégias já vistas anteriormente e aplicá-las.

Em nosso exemplo, como temos um relacionamento entre A e B de grau M:N, iremos derivá-lo segundo a regra que recomenda "criar-se uma tabela para representar o relacionamento R1 e estabelecer relacionamentos 1:N entre as TABELAS A-C e B-C". Se você voltar ao tópico que tratava esse assunto verá que a regra não era exatamente essa, mas que o resultado obtido pela migração de chaves estrangeiras acabava por gerar um modelo lógico com essa semântica.

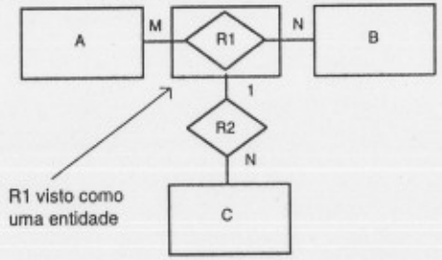
Assim, o modelo intermediário obtido pela derivação do aspecto relativo ao relacionamento será:



O aspecto interno de cada uma das tabelas geradas durante esse processo (A, B e R1), em termos de colunas, chaves primárias e chaves estrangeiras, será o mesmo já visto na derivação de um relacionamento M:N, por isso não iremos repeti-lo aqui. Caso você tenha dúvidas, consulte o tópico relativo a esse assunto.

2. Aspecto relativ[illegible] [illegible]ntidade

Após o primeiro passo (tratamento do aspecto relativo ao relacionamento), passaremos agora para o aspecto relativo à entidade. Isso significa que agora R1 deve ser visto no modelo conceitual como uma entidade.



Assim, deveremos procurar analisar o relacionamento de C em relação a R1, observando suas características (grau e presença ou não de atributos). Em nosso caso, vamos assumir que o relacionamento R2 (entre R1 e C) não possui atributos e que será derivado usando a regra básica para relacionamentos 1:N que define "migrar a chave primária de R1 para C".

Entretanto, quando falamos em chave primária de R1, devemos parar e responder a uma pergunta importante:

> *Quem é R1 neste ponto em nosso modelo?*
> *No que ele se transformou após o primeiro passo da derivação?*

Em nosso exemplo, que propositadamente foi construído para ser simples, R1 transformou-se em uma tabela que tem como chave primária as chaves estrangeiras de A e B que migraram para formar a chave primária de R1 (regra da derivação de relacionamentos M:N). Logo, se R1 existe concretamente como uma tabela, nosso modelo passa a ter o seguinte aspecto, após a derivação do aspecto relativo à entidade.

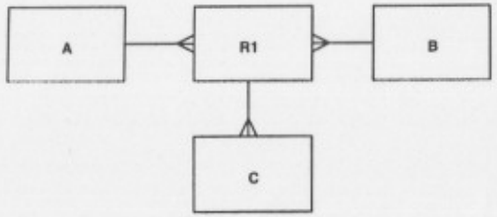


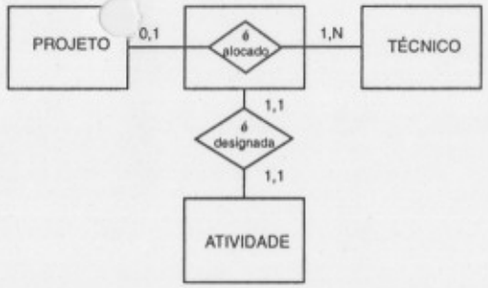
TABELA A	TABELA B	TABELA R1	TABELA C
coluna A1 (PK) coluna A2 (PK) coluna A3 coluna A4...	coluna B1(PK) coluna B2 coluna B3 coluna B4 coluna B5...	coluna A1(PK)(FK) coluna A2 (PK)(FK) coluna B1(PK)(FK)	coluna C1 (PK) coluna A1 (FK) coluna A2 (FK) coluna B1 (FK) coluna C2 coluna C3...

Mas o que teria acontecido se o relacionamento entre A e B fosse de grau 1:1 ou 1:N sem atributos? Sabemos que durante o processo de derivação de um tipo de relacionamento desse grau não é criada nenhuma tabela para preservar a existência de R1. Como então poderíamos relacionar R1 com C? Veja no exemplo a seguir um caso em que isso poderia ter ocorrido.

Exemplo:

Suponha uma empresa de prestação de serviços técnicos em projetos de construção civil. A cada projeto que se inicia, são estudadas as possíveis alternativas de alocação de recursos humanos e definida a equipe de trabalho. Em função das pessoas alocadas à equipe, são divididas as atividades a serem executadas durante o projeto. Isso é necessário, pois, a cada vez, diferentes técnicos com diferentes perfis de conhecimento são escolhidos. Assim, pode-se perceber que a designação das atividades para um técnico executar não se dá nem quando o projeto é criado e nem quando o técnico é contratado, mas sim quando o técnico "é alocado" a um projeto. Adicionalmente, um projeto contará com pelo menos um e até vários técnicos e um técnico poderá participar de nenhum, de um ou de vários projetos com diferentes atividades em cada um deles.

O modelo conceitual que representaria essa situação seria o seguinte:



Perceba que entre PROJETO e TÉCNICO temos um relacionamento de 1:N, que participa de uma agregação com ATIVIDADE. A cada alocação de um técnico a um projeto (aspecto do relacionamento na agregação), são associadas as diversas atividades que esse técnico deverá desempenhar no projeto. Conceitualmente nosso modelo consegue ser fiel a todas as condições estabelecidas no enunciado do exemplo. Isso demonstra que, apesar de pouco comum, a agregação pode ocorrer com relacionamentos de grau 1:1, 1:N e M:N e não somente nesse último, como alguns autores afirmam.

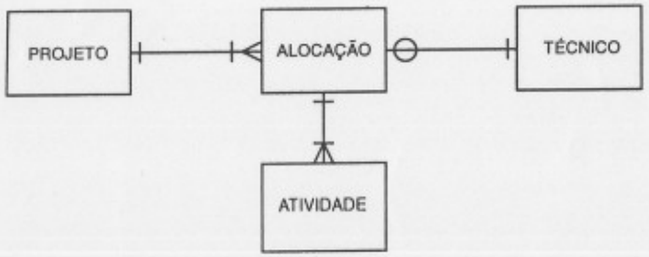
Mas voltando ao processo de derivação, que é nosso interesse maior, vejamos o que acontecerá na derivação do aspecto relativo ao relacionamento na agregação. Ao derivarmos o relacionamento entre PROJETO e TÉCNICO, obteremos a seguinte estrutura:

Teremos ainda, para derivar o aspecto relativo à entidade na agregação, que relacionar ATIVIDADE com a entidade representativa da agregação. Mas que entidade ou tabela é essa? Nesse caso, teremos duas alternativas clássicas:

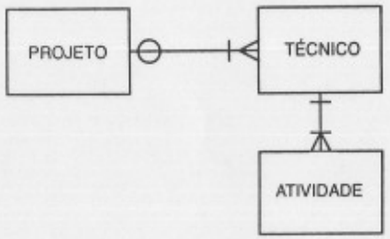
1. Criar uma tabela representativa do aspecto entidade na agregação (utilizando o critério de derivação de relacionamentos 1:N com atributos).
2. Relacionar a entidade ATIVIDADE com a entidade (tabela) TÉCNICO (entidade que aparece associada ao grau N do relacionamento).

No primeiro caso, obteríamos uma estrutura resultante muito mais próxima do modelo conceitual original. Isso garantiria, quando da utili-

zação do modelo lógico derivado, maior integridade no estabelecimento de relacionamentos válidos.



No segundo caso (que recomenda migrar o relacionamento com ATIVIDADE para TÉCNICO) obteremos uma estrutura com o seguinte aspecto:



Perceba que estaremos criando uma estrutura que, mesmo que venha a manter sua integridade referencial entre a tabela ATIVIDADE e TÉCNICO, deixará margens a falhas conceituais tais como:

1. Caso um TÉCNICO seja desvinculado de um PROJETO e não sejam tomadas as providências necessárias, poderemos ter, ainda, ATIVIDADES alocadas a esse TÉCNICO.
2. Se um TÉCNICO não possuir ATIVIDADES associadas a ele não será possível afirmar, com 100% de certeza, que ele não está alocado a nenhum PROJETO.

Esses tipos de falhas na manipulação da estrutura lógica, originadas pela adoção da segunda estratégia de derivação, não aconteceriam no modelo se a primeira estratégia fosse adotada, pois, mantida a integridade referencial (o que implica somente referências a valores de chaves primárias existentes), já teríamos garantida também a integridade conceitual.

Somente para que não tenhamos deixado de mencionar o processo de derivação de uma agregação quando o relacionamento na agregação é de

grau 1:1, iremos mencionar que as observações feitas para a derivação com relacionamento 1:N são todas válidas com relacionamentos 1:1. Adicionalmente, caso a segunda estratégia de derivação seja escolhida, a migração do relacionamento com a terceira entidade da agregação (ATIVIDADE no nosso exemplo) poderá ser feita tanto para uma como para outra entidade (PROJETO ou TÉCNICO).

5

Restrições no Modelo Lógico

Deixamos esse assunto para ser tratado após o termino do capítulo relativo ao processo de derivação do modelo lógico justamente por considerar que os tópicos a serem abordados aqui são complementares ao processo de derivação. Muitas vezes, em outras abordagens feitas por outros autores, eles poderão vir inseridos antes ou durante o próprio processo de modelagem lógica. Isto não torna a abordagem errada mas, teoricamente, insere alguma complexidade desnecessária naquele instante.

O assunto "restrição" será tratado sob três aspectos bastante distintos, mas que para efeito de compreensão foram agrupados em um único capítulo. Estes três aspectos serão:

- Restrições de Domínio.
- Restrições de Integridade.
- Restrições de Implementação.

O aspecto *restrições de domínio* diz respeito às considerações de implementação a serem avaliadas para cada uma das colunas, em função dos possíveis valores a serem assumidos em função de características do tipo de dado e do próprio SGBD onde essa coluna venha a ser alocada. Já o aspecto *restrições de integridade* diz respeito a uma questão tanto de implementação (características do SGBD) como lógica (de preservação de integridade das informações). E, por último, o aspecto *restrições de implementação* tem um caráter puramente restritivo de implementação sendo, portanto, dependente fortemente do SGBD escolhido. Veremos a seguir cada um desses itens em detalhes.

Restrições de Domínio

A preocupação com restrições de domínios, para colunas de uma tabela, é um trabalho final executado durante a fase de definição dessas tabelas através da linguagem DDL de um SGBD específico. Terá, portanto, as limitações desse próprio SGBD, apesar de, em sua grande maioria, ter características muito próximas uns dos outros em função do próprio padrão SQL já estabelecido e implementado, com pequenas variações, nos SGBDs comerciais.

O que determinará a restritividade (ou permissividade) na a uição de valores para uma coluna será, em princípio, um dos itens a seguir:

- **O tipo de dado da coluna.** O tipo de dado será identificado durante a fase de projeto lógico, ou mesmo, até prematuramente, durante o processo de modelagem conceitual. Basicamente ele será determinado pela próprio nome do item de dado. Por exemplo, um NOME DE PESSOA e uma QUANTIDADE DE PEÇAS terão domínios que deverão, respectivamente, ser especificados como "literal" e "número". Logo, para a coluna QUANTIDADE DE PEÇAS não será permitido o valor "ABC", o que já configura uma restritividade de domínio natural para o tipo de dado definido.

- **O tipo de representação interna escolhida para o dado.** Outro aspecto que impactará a restritividade de domínio para uma coluna será o tipo de representação interna escolhida para essa coluna. Vimos que para um campo do tipo "número" não poderíamos ter o valor "ABC". Mas que tipo de representação de números escolheremos? Números inteiros longos? Números inteiros curtos (limitados a 32.767)? Números com casas decimais? Números com precisão científica (muito grandes ou muito pequenos)? Se no caso do campo QUANTIDADE DE PEÇAS tivéssemos definido que seu tipo de representação seria um "inteiro curto", não poderíamos atribuir como valor válido 3,01. Esse tipo de restrição também poderá estar presente em outros tipos de dados como, por exemplo em literais curtos e literais longos, ou outros.

- **As características conceituais de presença ou não desse dado.** Durante o projeto conceitual de nosso modelo poderemos ter optado por algum tipo de abstração que nos tenha levado a obter entidades para as quais alguns atributos não terão valores estabelecidos em um dado instante (lembre-se do estudo dos valores "nulos inaplicáveis" e "nulos desconhecidos"). Este será um elemento determinante para que agora, durante o projeto de nossas tabelas, assinalemos as colunas como "permitidas" ou "não permitidas" para conter valores nulos. Se "proibirmos" a existência de valores nulos em uma coluna, estaremos restringindo o domínio que ela pode assumir. Também durante a derivação de nosso modelo lógico poderemos ter optado por algumas estratégias que tenham nos levado a obter tabelas onde teremos colunas que permitirão a existência de valores nulos. Este será mais um fator determinante de restrição de domínios em nossas colunas.

- **A condicionalidade de um relacionamento representado pela coluna.** Devemos lembrar que todo o relacionamento condicional, cujo grau mínimo de relacionamento seja zero, deverá permitir para as colunas nadoras da chave estrangeira desse relacionamento a existência de valores nulos (demonstrando a não "ligação entre as tabelas para essa linha"). Por conseguinte, os relacionamentos que não se enquadrem no caso anterior terão restritividade de valores nulos nessas colunas, tornando obrigatório seu preenchimento (ou estabelecimento do relacionamento).

- **A especificação explícita de intervalos para domínios contínuos.** Vimos, quando tratamos o assunto relativo a domínios, que um atributo poderia ter um domínio pertencente a um intervalo contínuo em função de sua própria característica no ambiente que está inserido. Isso significa, também, que a coluna que o irá implementar necessitará de controles ou especificações claras dos valores inicial e final do domínio a ser aceito. Isso determinará uma restritividade de valores para uma coluna advinda de restrições, não mais físicas, mas conceituais. Alguns SGBDs aceitarão essas especificações na própria construção das tabelas (projeto lógico) e outros necessitarão de meios "artificiais" para que essas restrições sejam impostas.

- **A especificação explícita de um conjunto de valores para domínios discretos.** A existência de valores predeterminados (coleção de valores) que poderão ser assumidos por um atributo deverá ser mapeada no projeto lógico, como uma restrição de domínio explícita. Também nesse caso alguns SGBDs terão características mais ou menos "amigáveis" para a implementação dessas restrições. Alguns permitirão que na própria especificação da tabela sejam associados à coluna em questão os valores permitidos. Outros, por sua vez, necessitarão de controles em nível de programação para que "em tempo de execução" sejam controladas as restrições necessárias.

- **A definição de alguns tipos especiais de dados.** A grande maioria dos gerenciadores relacionais já apresenta, como tipos de dados, alternativas baseadas em domínios bastante específicos e restritivos. Por exemplo, colunas que necessitem armazenar dados do tipo DATA, ou HORÁRIO, ou VALOR MONETÁRIO, etc. já podem ter, através da escolha de um tipo de dado especialmente construído para cada uma dessas situações, todas as restrições necessárias implementadas automaticamente. Assim será tão simples tornar proibitivo uma coluna numérica aceitar valores alfabéticos como fazer com que uma coluna do tipo DATA não aceite um valor igual a 31 de fevereiro de 1996. Alguns gerenciadores de bancos de dados orientados a objetos vão mais longe e permitem que se criem novos tipos de dados e domínios segundo as necessidades da aplicação.

Restrições de Integridade

O aspecto relativo à restrições de integridade não afetará diretamente a atribuição de valores isoladamente em uma linha de uma coluna. Normalmente, estará envolvendo diferentes linhas da mesma tabela, ou de tabelas diferentes. Em alguns casos terá a função de estabelecer (implementar) no modelo lógico algumas das restrições tecnológicas do modelo relacional e, outras vezes, simplesmente, terá a função de manter a integridade (fidelidade, correteza, coesão, etc.) entre os dados armazenados no banco de dados.

O modelo relacional, em função de seu método de estruturação e manuseio dos dados, depende, em grande escala, de estruturas lógicas íntegras. Isso significa que, se uma linha de uma tabela deve estar associada a outra linha de outra tabela, isso deverá ser garantido através dos dados contidos nessas tabelas. Se uma das linhas for "corrompida" por algum processo que as manuseia, poderemos estar "quebrando a integridade" entre essas tabelas. Para que isso não aconteça, os SGBDs devem (ou deveriam) implementar meios para que, durante a execução das diversas operações sobre os dados, não sejam "inseridos" pontos de "quebra de integridade".

Existem muitas discussões sobre quais SGBDs implementam efetivamente esses mecanismos de controle de integridade ou não. Muitos SGBDs ainda não atingiram o ponto de automação necessário e acabam por transferir a responsabilidade pela manutenção da integridade dos dados (entre tabelas) para os programas que manipulam suas estruturas. O ponto principal é que, automaticamente ou não, deveremos assegurar o grau mínimo de restrições de integridade para as estruturas de bancos de dados que estejamos criando durante nosso projeto lógico. Talvez isso venha até a implicar transferir para as aplicações algumas tarefas adicionais.

Dentre as restrições de integridade que devemos tratar estão:

- **A definição de unicidade de valores em certas colunas das tabelas.** Esse item é importante sob dois enfoques. Primeiramente, está associado de forma direta aos conceitos de chaves primárias de nossas tabelas. Vimos nos princípios da teoria relacional que uma tabela deve possuir um meio para assegurar que não existam duas linhas iguais. Vimos também que esse meio é a determinação de uma chave primária. Portanto, a chave primária (seja formada por uma ou mais colunas) não deverá permitir a ocorrência de valores duplicados. Se isso ocorrer, todo o processo relacional terá sido corrompido. Essa restrição é implementada pela totalidade dos SGBDs relacionais pois, caso não fosse, não teriam características efetivamente relacionais.

  O segundo enfoque desse item está associado às colunas não pertencentes à chave primária de uma tabela. Portanto, não tem envolvimento com aspectos de implementação tecnológica e sim com aspectos conceituais. Vamos imaginar uma tabela de AUTOMÓVEL que possua como chave primária o NÚMERO DO CHASSI, mas que possua também outras colunas tais como COR, MODELO, ANO DE FABRICAÇÃO, NÚMERO DA PLACA, etc. Poderíamos, dentro de nosso escopo de modelagem definir que, pelas regras conhecidas no ambiente modelado, não seriam permitidos NÚMEROS DE PLACAS iguais. Perceba que NÚMERO DA PLACA não é uma coluna da chave primária da tabela e, portanto, não teria qualquer restrição sobre a existência de valores repetidos, mas que, para fins de integridade conceitual, não deveremos permitir tais valores.

- **A restrição de referência de uma chave estrangeira a um valor de chave primária inexistente.** Por definição, uma chave estrangeira é uma chave primária de uma tabela migrada para outra tabela a fim de estabelecer vínculo com a primeira. Esse vínculo se estabelecerá através de referências de valores comuns entre as tabelas. Se na chave estrangeira de uma tabela tivermos valores que "apontem" para valores da chave primária inexistentes teremos uma estrutura "corrompida". Não conseguiremos estabelecer o relacionamento entre as tabelas, o que demonstra falta de integridade em nosso banco de dados.

TABELA PESSOA

cpf (PK)	nome
001	João da Silva
002	Pedro Paulo Sá
005	Marcos Silveira
010	Maria Tereza

referência inválida

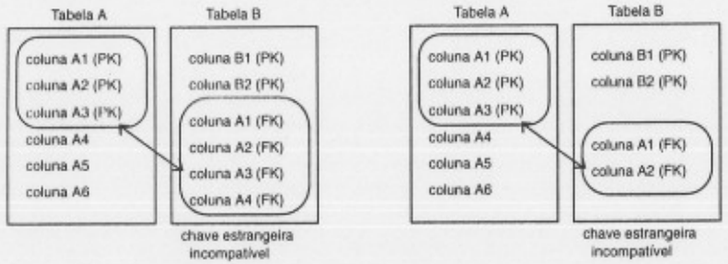
TABELA AUTOMÓVEL

num.chassi (PK)	cpf (FK)
00B125KN	010
34J556NS	002
993MO133	010
873F1345	003
48C446SW	001

Este será um tipo de restrição de integridade que não será verificado durante o processo de construção das tabelas (execução de comandos DDL) mas somente durante o "povoamento" das tabelas ou sua operação normal.

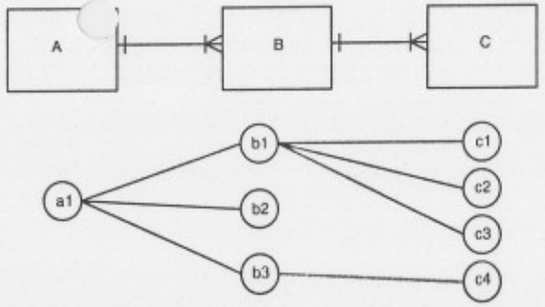
- **A restrição de referência de uma chave estrangeira a uma coluna, ou composição de colunas, de uma chave primária inexistente.** Essa restrição é verificada já durante o processo de definição das tabelas através da linguagem DDL, mais exatamente durante a execução desses comandos. Seremos alertados de erro na especificação de uma chave estrangeira se nossa especificação não for coincidente com a especificação da chave primária referenciada.

Caso isso não acontecesse, teríamos "perdido" parte da chave primária em nossa chave estrangeira, ou agregado novos dados a uma chave estrangeira que não pertenceriam à chave primária. Nesses dois casos, não conseguiríamos, como no item anterior, ser fiéis no estabelecimento de relacionamentos entre as tabelas.



- **A referência de uma chave estrangeira a uma tabela inexistente.** Também essa restrição é verificada durante o processo de execução dos comandos DDL de criação da chave estrangeira. Hoje, com a utilização, cada vez mais freqüente, de softwares de gerenciamento e de projeto de bancos de dados (interfaces de alto nível entre o DBA e o SGBD) tornam-se, cada vez menores, as possibilidades de ocorrências desses tipos de falhas. A exclusão de uma tabela que tenha sua chave principal sendo incorporada como chave estrangeira em outras tabelas também acusaria inconsistência caso essas referências não fossem excluídas sincronamente.

- **A exclusão de ocorrências (tuplas) em modo cascata em estruturas do tipo "hierarquia".** Caso tenhamos em nosso modelo estruturas do tipo "hierarquia" (o que é bastante comum), deveremos assegurar, através do SGBD ou via programação, que a exclusão de um "pai" na estrutura, reflita, quando necessário, a exclusão dos "filhos" e assim por diante. Em alguns casos, quando o relacionamento for condicional (por exemplo 0,1:1,N) poderemos ter situações onde os "filhos" venham a ser preservados mesmo após a exclusão do "pai". Deveremos analisar caso a caso e verificar qual é a "cadeia de exclusão" necessária. Caso não tenhamos assegurada essa restrição de integridade poderemos ter, após a exclusão de um determinado "pai", filhos referenciando-o, o que denotará claramente uma inconsistência na base de dados.



Se nessa estrutura, excluíssemos a ocorrência **a1**, deveríamos assegurar que "em cascata" as ocorrências **b1**, **b2** e **b3** também fossem excluídas e que, por sua vez, **b1** sendo excluído também **c1**, **c2** e **c3** o fossem. Como **b3** foi excluído pela exclusão de **a1**, então **c4** também deveria ser excluído. Isso manteria a "integridade" de nosso modelo.

- **A alteração de ocorrências (tuplas) em modo cascata em estruturas do tipo "hierarquia".** Assim como nas exclusões "em cascata", deveríamos repassar para os elementos participantes da hierarquia os efeitos de operações realizadas nos mais altos níveis, também nas alterações deveremos ter essa preocupação. Quando, por exemplo, alterarmos um valor de uma coluna em uma linha, e esse valor aparecer "espelhado" em outras linhas de outras tabelas, seja como chave estrangeira, ou como dependência de identificador, deveremos repassar esse novo valor para todos os pontos onde ele apareça, sob o risco de, caso não completemos essa operação, perder a integridade em nosso banco de dados.

- **A manutenção de sincronismo entre a tabela e os índices da tabela.** Alguns SGBDs, para fins de melhoria de desempenho durante processamentos de grandes quantidades de atualizações (inserções, exclusões ou alterações), permitem que as estruturas de índice (normalmente implementadas como árvores binárias balanceadas) sejam mantidas off-line. Com isso, enquanto as tabelas estão sofrendo grandes volumes de atualização, suas estruturas de índices não são atualizadas sincronamente. Isso faz com que não seja necessário "manter balanceada" a árvore de índices, o que agiliza muito o processo de atualização. Entretanto, deverá existir um momento (após o término da atualização maciça) em que a estrutura de índice seja totalmente reconstruída para espelhar a nova tabela já atualizada. Enquanto isso não for feito estaremos manuseando uma estrutura "não-íntegra" de índices, o que poderá ter efeitos imprevistos.

Restrições de Implementação

O tratamento das restrições de implementação deverá ocorrer já durante a fase de derivação do modelo lógico. As decisões sobre quais estratégias utilizar durante o processo de derivação de relacionamentos e das estruturas de agregação e generalização-especialização dependerão das restrições de implementação impostas pelo SGBD que se esteja utilizando. Assim é importante que, antes de se iniciar a derivação dessas estruturas, sejam obtidas informações sobre quais as características do SGBD utilizado.

Algumas dessas restrições poderão ser também impostas não pelo SGBD mas por ferramentas CASE que estejam sendo utilizadas durante a modelagem conceitual e lógica das bases de dados. Muitas vezes, por restritividade de representação ou mapeamento na ferramenta CASE, acabaremos sendo restringidos de usar este ou aquele recurso do SGBD mesmo que ele próprio não os limite. Dentre as restrições de implementação que possivelmente serão encontradas estão:

- **A existência de itens de repetição (vetores uni ou n-dimensionais).** Essa restrição não é conseqüência da inabilidade de implementação de um ou outro fornecedor. Ela é, na verdade, reflexo da maior ou menor aderência às formas normais (neste caso a 1FN), pois se lembrarmos bem a 1FN diz que "não deverão existir itens de repetição...".
  A observância da 1FN, durante o processo de normalização, impõe a restrição do uso de estruturas de repetição, pois o próprio SQL não está preparado para manuseá-las. Entretanto, alguns fornecedores, tentando incorporar algumas habilidades adicionais a seus produtos, podem "disponibilizar" esse tipo de estrutura. Não deverá ser um recurso encontrado com muita facilidade pois "desafia" os preceitos da própria teoria relacional. Mesmo que seja encontrado em algum produto, deveria ser visto como algo desaconselhável, pois restringirá futuramente qualquer processo de migração ou integração com outros SGBDs e ferramentas de acesso.

- **Implementação direta ou manuseio transparente de estruturas de generalização-especialização.** Vimos alguns meios para derivar as estruturas de especialização-generalização. Todos eles partiam do pressuposto de que as estruturas não poderiam ser manuseadas "naturalmente" por um SGBD relacional. Esta, infelizmente, é a realidade atual. Somente os SGBDs orientados a objetos implementam, hoje, o armazenamento e manuseio transparente dessas estruturas. Desse modo, se você estiver utilizando um SGBD relacional, pode considerar essa restrição como aplicável ao seu ambiente. Você deverá, através das estratégias de derivação apresentadas, criar meios para o armazenamento e a manipulação dessas estruturas.

Dentro da [illegible]va padronização da linguagem SQL, atualmente em execução (o SQL-3), já estão sendo incorporadas novas habilidades para o tratamento dessas estruturas e de tantas outras características oriundas do ambiente de orientação a objetos. Com isso, em um futuro próximo, não mais teremos essa restrição no ambiente relacional.

6

O Ambiente Corporativo

A partir deste ponto, iremos abordar alguns aspectos relativos à modelagem em ambientes corporativos integrados. Nesses ambientes, dificuldades adicionais poderão surgir em função da extensão e diversidade de elementos a serem modelados. Diversas equipes, em diferentes áreas, poderão estar sincrona, ou assincronamente, identificando e elaborando modelos complementares ou até mesmo redundantes. Fazer com que os resultados individuais produzidos em cada modelo possam representar, efetivamente, um somatório de resultados para a corporação exigirá meios para que os esforços sejam conduzidos sempre em um mesmo sentido.

Obtendo o Modelo Corporativo

Como vimos anteriormente, a modelagem de dados se aplica à criação de modelos nos mais diversos níveis de detalhe e abrangência possíveis. Dependendo das especificações de requisitos feitas para a elaboração desses modelos, poderemos tanto abordar pequenos segmentos de um universo como todo o universo em questão.

Essa especificação, portanto, pode, em alguns casos, definir que a abrangência desejada para o modelo de dados a ser gerado seja aquela relativa a toda a organização ou empresa. Nesse caso, estaremos buscando o que se define como *modelo corporativo*. Esse modelo, em princípio, contém a representação de todos os objetos, ou dados, de interesse de uma organização.

Sim, mas como obter esse modelo se sabemos que teremos um universo bastante grande de dados a levantar, analisar e modelar? Como viabilizar a execução de um trabalho dessa extensão? Como garantir que teremos abrangido todo o universo desejado ao fim de nosso trabalho? Essas perguntas têm sido bastante discutidas, e basicamente três formas de tratamento podem ser dadas para que se atinjam os resultados desejados. Dentre essas três formas encontramos, basicamente, uma abordagem de análise, que parte do geral para o específico, uma abordagem de síntese, que parte do específico atingindo o geral, e uma abordagem combinatória que busca a combinação das duas anteriores.

A Abordagem Top-down

A primeira forma possível para se atingir o modelo corporativo, defendida por alguns especialistas e profissionais ligados à área de modelagem, é aquela em que se define que o modelo deve ser buscado através de um trabalho que parta do geral para o específico, ou seja, do conhecimento do todo para o conhecimento das partes.

Através dessa abordagem, teoricamente, pode se ter maior controle sobre o atendimento do critério abrangência, assegurando que o ambiente observado seja tratado em sua totalidade sem esquecimento de algum de seus componentes. Neste método, deveremos, então, procurar ter uma visão geral da corporação, modelar seus macroobjetos, e depois então procurar sucessivamente buscar o detalhamento desses macroobjetos em "n" níveis. A cada nível de detalhamento em que se aumenta a profundidade da análise, aumenta-se, também, a amplitude de objetos tratados.

Usando uma analogia, fora da área de informática, para ilustrar esse tipo de abordagem, poderíamos imaginar o caso em que uma expedição exploratória deseja fazer o mapeamento de todas as espécies de plantas existentes em uma floresta desconhecida, identificando-as detalhadamente, região por região.

Na abordagem top-down a equipe deveria, em um primeiro instante, através de mapas ou sobrevoando a área a uma altitude média, buscar uma visão bastante abrangente da floresta, para poder identificar as regiões que terá que analisar futuramente. Só assim ela será capaz de identificar o relevo, as delimitações da floresta, os acidentes geográficos nela contidos (lagos, rios, montanhas, etc.) e outros elementos que dêem noção da extensão do trabalho a ser executado, dos recursos a serem aplicados e das dificuldades a serem transpostas.

Esse procedimento inicial seria interessante pois, caso a equipe decidisse se embrenhar na floresta sem esse conhecimento prévio, provavelmente ficaria perdida ou teria grandes dificuldades em completar o trabalho. Por quê? Porque sua tendência seria a de fixar-se ao detalhamento dos primeiros espécimes identificados já na primeira região. Isso talvez trouxesse resultados imediatos quanto ao mapeamento da primeira região, mas impediria a geração de resultados abrangentes em um curto espaço de tempo. Ou seja, passariam-se meses e meses de trabalho e não poderíamos sequer estimar o percentual de trabalho já concluído, pois nunca teríamos tido, até o momento, a visão da extensão total da floresta. Tanto ela poderia ser pequena e o trabalho estar em vias de ser acabado, como ela ser gigantesca e o trabalho representar não mais que 0,5% do total a ser realizado. Como saber se a floresta já havia sido toda explorada? Como saber se deveríamos aumentar a equipe de exploração? Como distribuir os recursos disponíveis? Seriam sempre incógnitas.

Transportando esse exemplo para a área de modelagem de dados, veremos que também assim deve ser visto o trabalho de construção de um

modelo corporati Desejamos o mapeamento de toda a corporação, e o conhecimento de cada uma das áreas e assim deveremos traçar uma estratégia de atuação que nos garanta esse resultado.

Alguns outros pontos também devem ser mantidos em mente. O primeiro deles é que nossa tarefa deve ser abrangente. Não nos interessa o detalhamento de uma área e a total ignorância de outras "ainda não exploradas". Devemos conseguir, em um espaço de tempo "razoável", um mapeamento que seja homogêneo para toda a corporação, sem que esta ou aquela área deixe de ser abordada. Precisamos conhecer as dimensões do trabalho a ser executado para dimensionar nossos recursos. Precisamos conhecer características internas que nos guiem durante a execução dos trabalhos.

Como pode ser percebido, o conhecimento do todo, em um instante inicial, é bastante importante para que sejamos orientados durante as próximas fases de detalhamento dos modelos. Isso não significa, entretanto, que somente a aplicação da abordagem top-down seja suficiente. Muitos projetos têm falhado na concepção do modelo corporativo por buscarem sua construção através de uma abordagem top-down equivocada.

A abordagem top-down deve procurar ignorar o maior número possível de detalhes nos níveis mais altos do modelo. Tentar representar um modelo corporativo trabalhando em níveis de detalhe excessivamente grande seria a mesma coisa que tentar desenhar o mapa do Brasil com todas as suas estradas, ruas, avenidas, quadras, lotes e casas. Se você deseja ter a visão geral do Brasil, construa um mapa somente com as principais rodovias. Se você deseja detalhes de urbanização de cada uma das cidades faça o mapa detalhado dessas cidades!

Logo, a simples união dos diversos modelos das áreas de uma corporação não resultará em um modelo corporativo! Agregando esses modelos teremos uma agregação de objetos que tornará impossível a compreensão do todo, indo contra as principais características esperadas de um bom modelo: concisão, precisão, objetividade e clareza. Esta tem sido uma das grandes falhas em alguns processos de obtenção de modelos corporativos. Se seu modelo corporativo não for sucinto o suficiente para dar uma visão do todo, então você não terá atingido o ponto ideal. Lembre-se: cada mapa deve ter somente o detalhamento suficiente para o tornar útil e inteligível, e nada mais que isto!

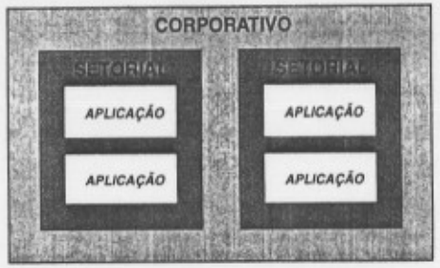
Não esqueça, porém, que somente a criação dos mapas das cidades ou somente a criação do mapa do Brasil não resolverá nosso problema de mapeamento. Deveremos construir, necessariamente, vários mapas, cada um com seu nível de detalhamento possível, e com sua aplicação específica. Assim também com relação aos modelos de dados, deveremos gerar "n" níveis de representação do nosso ambiente observado. A esses níveis chamaremos de corporativo, setorial e de aplicação. Veremos maiores detalhes sobre eles mais a frente.

A Abordagem Bottom-up

A abordagem bottom-up define, por outro lado, que o ponto de partida para a obtenção do modelo corporativo deve ser o conhecimento das partes que o compõem, para que então, pela associação dessas partes, possa-se chegar ao todo. Esse raciocínio baseia-se no fato de que as pessoas têm uma melhor visão e entendimento de áreas operacionais restritas. Isso porque seus focos de atenção estarão voltados para as áreas nas quais já estão inseridas ou das quais já participam.

Sem dúvida esses argumentos são bastante válidos se desejarmos obter maiores detalhamentos para nosso processo de modelagem. A busca dos objetos existentes em uma área de nosso interesse para modelagem acabará por ser feita através de um processo bottom-up. Será desse modo, predominantemente, que nos serão apresentadas as informações quando estivermos executando o processo de levantamento de dados. Entretanto, ao se gerarem diversos modelos de áreas operacionais (bottom-up), estarão sendo criadas "fronteiras" e "interseções" que deverão ser tratadas.

Justamente nesse ponto é que entrarão as habilidades necessárias à integração dos modelos de dados obtidos. Nesse instante, passamos a ter de lidar com alguns novos conceitos: os modelos de dados de aplicação e os modelos de dados setoriais. Veremos o que significa cada um deles.



Modelo de Dados de Aplicação

Esse termo, que também pode ser referenciado como *modelo de dados de sistema*, ou *modelo de dados local*, aplica-se ao resultado do processo de modelagem que é executado dentro de um escopo restrito: o de um sistema ou aplicação. Quando falamos em sistema, ou aplicação, estamos nos referindo ao conjunto de rotinas sistematizadas que, de modo completo, dá atendimento a uma área funcional da organização. Assim, um sistema de folha de pagamento ou uma aplicação de controle de fornecedores poderia exemplificar esse conceito.

Imaginemos agora que estamos realizando um processo de modelagem de dados que pretende abranger esses dois sistemas citados. Apesar

de estarmos focalizando o ambiente de uma mesma organização, as áreas funcionais (veja que não estamos nos referindo a áreas de um organograma!) por eles envolvidas são distintas. Deveríamos então executar um modelo para cada uma dessas aplicações, assim diríamos ter o Modelo de Dados em Nível de Aplicação da Folha de Pagamento e do Controle de Fornecedores.

Por que se justifica esse tipo de abordagem? Porque em última análise esse é o nível operacional onde realmente estão envolvidos os objetos que nos interessam para a modelagem de dados. É nesse nível que obteremos os detalhes de caracterização e de relacionamento entre os objetos. É nesse nível que futuramente teremos a implementação de possíveis estruturas de dados, visando à criação de sistemas informatizados ou de bancos de dados. E, finalmente, porque é nesse nível que as pessoas estão inseridas durante a execução de suas atividades diárias, fazendo que elas então possam compreender e nos repassar os conceitos necessários.

Imagine agora que essa abordagem será válida para cada uma das áreas funcionais da organização. Que cada aplicação, ou sistema, será modelado no nível operacional em que está inserido. O que teremos ao final de algum tempo de trabalho? O modelo corporativo? Não! Como já dissemos, a união das partes nem sempre leva ao todo! Pelo menos não a um todo compreensível.

Surge assim a necessidade de um trabalho de integração dos modelos de aplicação. Esse trabalho culminará em um modelo que chamamos de *modelo de dados setorial*.

Modelo de Dados Setorial

Esse termo, que também pode ser referenciado como *modelo de dados departamental* ou *modelo de dados de assunto*, aplica-se ao resultado do processo de modelagem de dados que é executado dentro de um escopo já não tão restritivo como o do *modelo de dados de aplicação*.

Estaremos trabalhando, neste modelo, restritos a um setor funcional, ou departamento, da organização, procurando agrupar os modelos de dados de aplicação cujas funcionalidades digam respeito a uma mesma área de atuação ou assunto. Por isso, o termo *setorial* é mais adequado que o termo *departamental*, pois esse último traz influências de aspectos formais do organograma da empresa, o que, nesse caso, não nos interessa. Muitas vezes o uso do termo departamental induz à criação de um modelo orientado a uma área formal do organograma da empresa. Em alguns casos, essa área formal pode estar executando funções diversas, não agrupáveis, ou possuir em sua estrutura disfunções que levarão a um modelo de dados departamental não coerente.

O modelo de dados setorial será obtido pela integração dos modelos de dados de aplicação que focalizem um mesmo assunto. Entretanto, esse

modelo não será simplesmente a junção dos objetos existentes nos modelos de dados de aplicação. Teremos dois trabalhos distintos a serem executados durante sua elaboração: a compatibilização e o nivelamento.

Modelo de Dados Corporativo

Nesse último nível, após a compatibilização e nivelamento dos modelos de aplicação e obtenção dos modelos setoriais, daremos início a um novo processo de integração. Todas as estratégias e técnicas utilizadas, até então, para a integração de modelos de aplicação, deverão ser utilizadas para a integração dos modelos setoriais. Assim, pelo agrupamento dos modelos setoriais, estaremos dando origem a um modelo global que espelhará o mapeamento de toda a corporação. A este modelo global denominaremos *modelo corporativo*.

Veremos durante a discussão do processo de compatibilização e nivelamento de modelos que cuidados especiais deverão ser tomados na construção do modelo corporativo. O simples agrupamento dos modelos setoriais, como muitos acreditam e praticam, não dará origem a um modelo corporativo efetivo. Técnicas especiais devem ser aplicadas para que esse processo produza um modelo manuseável e inteligível sob o ponto de vista semântico e operacional.

Compatibilização dos Modelos de Dados

Durante o processo de junção de dois ou mais modelos de aplicação para a obtenção de um único modelo de dados setorial, nos defrontaremos com quatro situações bastante típicas, que são:

Objetos	Caracterizações
IGUAIS	IGUAIS
DIFERENTES	DIFERENTES
IGUAIS	DIFERENTES
DIFERENTES	IGUAIS

Nas duas primeiras situações, não encontraremos dificuldades para o processo de compatibilização, pois um objeto X caracterizado como X em duas aplicações simplesmente deverá ser unificado como X, e um outro objeto Y caracterizado como Y, também em duas aplicações, do mesmo modo será unificado como Y.

Mas, e no caso de encontrarmos objetos iguais com caracterizações diferentes, como tratá-los? Assim, se W é caracterizado como W na aplicação 1 mas como M na aplicação 2, teremos que de algum modo descobrir que M na aplicação 2 é na verdade W. Pior que isso talvez seja o caso onde

Z é caracterizado na aplicação 1 como Z e Q é caracterizado na aplicação 2 como Z. Teremos dois objetos distintos sendo falsamente caracterizados como um único objeto, o que nem sempre é de fácil identificação.

Objeto Observado	Identificação na Aplicação 1	Identificação na Aplicação 2
X	X	X
Y	Y	Y
W	W	M
Z	Z	-
Q	-	Z

Exemplo:

Para melhor caracterizar a situação anteriormente apresentada, imagine que duas aplicações desenvolveram seus modelos e que o ambiente observado em cada uma delas é o seguinte:

- **Aplicação 1:**

Ambiente Observado: **Área administrativa da empresa**
Objetos Observados: **Fornecedor, Cliente, Funcionário, Órgão**

- **Aplicação 2:**

Ambiente Observado: **Área jurídica da empresa**
Objetos Observados: **Órgão, Norma, Advogado, Vara Cível**

A compatibilização desses dois modelos visando a um modelo corporativo poderia nos induzir à existência dos seguintes objetos corporativos:

Fornecedor, Cliente, Funcionário, Órgão, Norma, Advogado, Vara Cível

Entretanto, durante a análise dos objetos feita no processo de compatibilização, poderemos descobrir que:

- **Órgão** na **aplicação 1** diz respeito a elementos formadores da estrutura do organograma da empresa, tais como diretorias, departamentos, divisões, seções.
- **Órgão** na **aplicação 2** diz respeito às instituições normalizadoras existentes no nível municipal, estadual e federal, tais como o Conselho Regional de Medicina (CRM), Organização dos Advogados do Brasil (OAB), entre outros.
- **Funcionário** na **aplicação 1** diz respeito a todos os indivíduos que contratados pela empresa prestam serviços internos ou externos.

- **Advogado** na **aplicação 2** diz respeito a todos os indivíduos que, formados em Direito, foram contratados pela empresa para prestar serviços de orientação e assessoria jurídica.

O que podemos constatar com essas informações obtidas?

1. **Órgão** na **aplicação 1** e **Órgão** na **aplicação 2**, apesar de terem a mesma identificação, não são os mesmos objetos e, portanto, devem ser diferenciados.

2. **Funcionário** na **aplicação 1** e **Advogado** na **aplicação 2**, apesar de terem identificações diferentes, são o mesmo objeto visto por dois enfoques especializados mas que têm total harmonia, portanto, podendo ser integrados como uma só coisa.

Fica claro, com esses exemplos, que o processo de integração e compatibilização envolverá senso crítico, investigação, análise e reconhecimento dos objetos existentes nos modelos de dados das aplicações. Neste ponto, muita habilidade pessoal será exigida e de quase nada adiantarão os supostos softwares modeladores automáticos de dados.

Até o momento, estávamos considerando, para facilitar o entendimento, somente discrepâncias existentes entre uma aplicação 1 e uma aplicação 2. Vamos considerar agora que, no mesmo caso, já temos um modelo de dados setorial previamente criado e estamos tentando compatibilizar dois novos modelos de dados de aplicação com este já existente. Teremos aí uma situação onde, além de discrepâncias entre os modelos de dados de aplicações, poderemos ter discrepâncias entre estes e o modelo de dados setorial já concebido. Teremos de tratá-las de igual modo considerando, além disso, que conceitos e caracterizações previamente obtidos no modelo de dados setorial estejam incorretos. Isso implicará mudanças, ou correções dos modelos, tanto em nível setorial como em de aplicações.

Toda e qualquer alteração de um modelo de dados setorial deverá ser analisada sob o ponto de vista de impactos sobre os modelos de dados de aplicação que lhes deram origem. Essas alterações devem ser repassadas, quando necessário, para os demais modelos.

Todo o processo aqui descrito para sintetização do modelo de dados setorial a partir dos modelos de dados de aplicação é valido para a obtenção do modelo de dados corporativo a partir dos modelos de dados setoriais. Basta que se repita o processo com outro nível de agregação.

Nivelamento dos Modelos de Dados

Dentro do processo de obtenção de um modelo de dados corporativo a partir da abordagem bottom-up, deve-se aplicar o conceito de nivelamento de modelos para que se atinjam os resultados esperados.

Já citamos anteriormente que um modelo deve conter somente os elementos úteis a sua compreensão. Este, portanto, deve ser o princípio utilizado durante as três fases de modelagem:

Fase da Modelagem	Nível de Detalhamento
Modelagem da Aplicação	Grande
Modelagem Setorial	Médio
Modelagem Corporativa	Pequeno

Isso garantirá que nosso foco de atenção esteja voltado nas fases corretas para os objetos corretos. Lembre-se: podemos, e devemos, ter mapas das cidades, dos estados e do Brasil, mas nem todos eles com os mesmos níveis de informação.

Logo, caso estejamos integrando modelos de aplicação para obter um modelo setorial, deveremos estar atentos a elementos que tenham puramente função normalizadora ou detalhadora de objetos em nossos modelos. Esses elementos, durante a fase de nivelamento, deverão ser omitidos, ou englobados por elementos de nível de abrangência maior. Esse mesmo tipo de estratégia deverá ser adotado na obtenção do modelo corporativo a partir dos modelos setoriais fazendo com que, ao final do nivelamento, somente os elementos essenciais à corporação sejam mapeados e representados.

Exemplo:

No modelo da aplicação Recrutamento e Seleção de Funcionários poderíamos ter os seguintes elementos identificados:

- CANDIDATO
- DEPENDENTE
- DOCUMENTO PESSOAL
- ESPECIALIDADE
- VAGA

Ao buscarmos os elementos básicos para o nivelamento desse modelo, durante a integração com o modelo de Folha de Pagamento, poderíamos descartar na geração do modelo setorial de Administração de Recursos Humanos os elementos

- DOCUMENTO PESSOAL
- ESPECIALIDADE

ficando somente com os elementos essenciais que são:

- CANDIDATO
- DEPENDENTE
- VAGA

Nesse caso, os elementos descartados (DOCUMENTO PESSOAL, ESPECIALIDADE) foram englobados pelo elemento CANDIDATO. Desta mesma maneira, no nivelamento do modelo setorial de Administração de Recursos Humanos com outros modelos setoriais poderíamos, ainda mais uma vez, proceder à busca de elementos a serem omitidos no modelo corporativo.

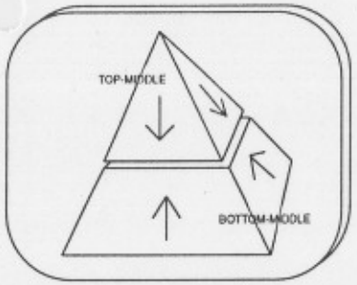
Em nosso exemplo, durante a fase de obtenção do modelo corporativo, provavelmente, acabaríamos por omitir CANDIDATO, DEPENDENTE e VAGA, que, do ponto de vista de recursos humanos, são bastante importantes, mas que para a empresa como um todo não dizem respeito. Assim, de nossa lista inicial de elementos acabaríamos por não manter no modelo corporativo nenhum dos elementos citados. Com certeza teríamos no modelo corporativo um elemento denominado FUNCIONÁRIO, ou EMPREGADO, que estaria representando a força de trabalho efetiva da corporação e que, em algum momento, passou pela condição de CANDIDATO. Entretanto, manter no modelo corporativo todo o ciclo de vida de recrutamento, seleção e administração de pessoal seria extremamente detalhista para a finalidade desejada.

A Abordagem Top-down x Bottom-up

Pode-se perceber, claramente, que cada uma das duas possíveis abordagens utilizadas para a obtenção do modelo de dados corporativo tem suas vantagens e desvantagens. Logo, qual deveria ser a melhor opção?

Como em tantos outros dilemas a serem enfrentados e resolvidos no dia-a-dia, devemos tentar, nesse caso, aplicar o princípio do "melhor dos dois mundos", ou seja, tentar buscar o que cada uma das abordagens oferece de melhor e conjugá-las. Acabaremos assim por estabelecer uma abordagem que chamaremos de *top-middle-bottom-middle*.

Nessa abordagem, como o próprio nome define, tentaremos conhecer e modelar os objetos de nosso ambiente a partir de uma visão geral, abrangente e pouco detalhada (obtida pelo processo top-down) mas não desceremos em seu detalhamento mais que o suficiente para conhecer poucos detalhes do ambiente a ser modelado. Isso fará com que capturemos os elementos suficientes para nos guiarmos no restante do processo e garantirá (pela visão top-down) que não haja omissão, ou desconhecimento, de áreas a serem exploradas no restante do trabalho.



Tendo obtido o modelo da fase top-middle, manteremos como nova referência esse modelo e partiremos para um novo trabalho, agora bottom-middle, ou seja, iniciando um processo que busque modelos detalhados, completos e com escopos limitados (obtidos pelo processo bottom-up). Com isso obteremos os modelos no grau de detalhamento necessário tendo, paralelamente, o referencial do modelo top-middle gerado no passo anterior. A partir do instante em que consigamos pelos modelos bottom-middle cobrir todos os aspectos relacionados no modelo top-middle, teremos atingido a completeza de nosso trabalho.

E Finalmente...

Esperamos ter traçado um caminho bastante claro e objetivo sobre o processo de modelagem conceitual e lógica. Utilizamos todas as estratégias que a experiência tem nos demonstrado serem úteis no processo de aprendizado de modelagem. Ordenamos e nivelamos o conhecimento em cada um dos capítulos cuidadosamente, sempre dentro de um plano que procurou apresentar somente o necessário a cada instante.

Gostaríamos de receber qualquer crítica ou sugestão para que melhorias possam ser inseridas nas próximas edições ou em novos materiais a serem publicados no futuro. Estamos abertos a trocar idéias e discutir pontos de vista sobre quaisquer dos pontos aqui apresentados.

Esperamos seu contato por qualquer um dos meios listados a seguir:

Correspondência para:	Tree Tools Informática Ltda. A/C Paulo Cougo Rua Camões, 1275, Curitiba, PR 80040-180
Contato telefônico em:	(041) 264-7205
E-mail para:	p.cougo@treetools.com.br
Consulta em:	http://www.treetools.com.br

Bibliografia

Barbieri, Carlos. *Modelagem de Dados*. Rio de Janeiro: Infobook, 1994.

Cerícola, Osvaldo Vicente. *Banco de Dados Relacional e Distribuído*. Rio de Janeiro: LTC - Livros Técnicos e Científicos Ed., 1991.

Chen, Peter. *Modelagem de Dados*. São Paulo: McGraw-Hill, Makron, 1990.

Chu, Shao Yong. *Banco de Dados: Organização, Sistemas, Administração*. São Paulo: Atlas, 1983.

Coad, Peter. *Análise Baseada em Objetos*. Rio de Janeiro: Campus, 1992.

Date, C.J. *Introdução a Sistemas de Bancos de Dados*. Rio de Janeiro: Campus, 1991.

Feliciano Neto, Acácio. *Engenharia da Informação: Metodologias, Técnicas e Ferramentas*. São Paulo: McGraw-Hill, 1988.

Korth, Henry F. *Sistema de Banco de Dados*. São Paulo: McGraw-Hill, 1989.

Machado, Felipe N. R. *Projeto de Banco de Dados : uma visão prática*. São Paulo: Érica, 1995.

Martin, James. *Análise e Projeto Orientados a Objeto*. São Paulo: Makron Books, 1995.

Martin, James. *Engenharia da Informação: Introdução*. Rio de Janeiro: Campus, 1991.

Martin, James. *Princípios de Análise e Projeto Baseados em Objetos*. Rio de Janeiro: Campus, 1994.

Martin, James. *Técnicas Estruturadas e CASE*. São Paulo: McGraw-Hill, 1991.

Rumbaugh, James *et al*. *Modelagem e Projetos Baseados em Objetos*. Rio de Janeiro: Campus, 1994.

Setzer, Valdemar W. *Bancos de Dados*. São Paulo: Editora Edgard Blücher Ltda., 1989.

Shlaer, Sally. *Análise de Sistemas Orientada para Objetos*. São Paulo: McGraw-Hill, 1990.

Índice

abordagens
bottom-up e top-down, 274
bottom-up, 268
top-down, 266
administração da complexidade, princípios da, 2
agregação, estrutura de, 131, 150
ambiente corporativo, 265
Análise Baseada em Objetos (Coad & Yourdon), 1, 5
ANSI-X3-SPARK, 25, 31
arquitetura em três níveis
integração com a abordagem E-R, 31
atributos
classificação quanto à finalidade, 60
descritivos, 60-61
nominativos, 61-63
referenciais, 63-64
no modelo conceitual, 63
dicionarização, 65
papel, 57

caracterização semântica
elementos adicionais, 108
apresentação de subgrupos, 124
especializações mutuamente exclusivas, 109
especializações não-mutuamente exclusivas (papéis), 114
estruturas de generalização e especialização, 108
estrutura Gen-Espec em auto-relacionamentos, 129
quando utilizar, 121-123
níveis de especialização, 116
CASE, ferramentas, 28-29
Chen, Peter P., 1
classes, conceito de, 3, 35
comunicação, instrumento de, 21
conceitos básicos, 7

dados
execução da modelagem de, 13-18
conceitos, entendimento de, 14
fidelidade e coerência, verificação de, 16
objetos, observação de, 14
objetos, representação de, 15
validação, 16
modelagem e orientação a objetos, 1
modelagem e projeto de bancos de dados, 4
objetivo do modelo de, 18
referencial para geração de estruturas de, 23
derivação
agregações, 247
aspecto relativo a entidades, 249
aspecto relativo ao relacionamento, 248
auto-relacionamentos, 231
1:1, 232
1:N, 233
M:N, 235
estruturas de agregação e generalização-especialização, 172, 237
atributos e relacionamentos especializados, 241
entidades generalizadas e entidades especializadas, 238
migração de atributos, 246
tabelas para entidades especializadas, 243
normalização das estruturas de dados, 172
regras, 171
relacionamentos, 172
DFD (diagrama de fluxo de dados), 22

dicionarização de objetos
regras básicas (entidades/classes), 52
requisitos a serem atendidos, 53
DOS, 30
DTE (diagrama de transição de estado), 22

elementos
observados, 3
reconhecimento no ambiente observado, 34
entidade-relacionamento, abordagem, 1, 28
entidades fortes e fracas
considerações, 53, 55
dependência e existência, 53
dependência de identificador, 55
entidades, conceito de, 3, 35, 37-46
coisas tangíveis, 38
especificações, 43
eventos ou ocorrências, 41
funções, 39
iterações, 42
objetos-especificação, 44
E-R, modelo, 33
abordagem entidade-relacionamento, 33
embasamento, 33
estruturado, processo de modelagem, 10-12

FORTRAN, 22

generalização-especialização, estrutura de, 150

ISO-9000, 22

Lei do Mundo, 34

modelagem
abrangência, 12
níveis
definição, 25
nível de detalhamento, 12
perspectivas da, 5
processo de, 12
recursos disponíveis, 13
tempo para a produção do modelo,13
Modelo Conceitual de Dados (MCD), 28-29
modelo de dados
compatibilização, 270
de aplicação, 268
de sistema, 268
local, 268
nivelamento, 272
setorial, 269
Modelo Físico de Dados (MFD), 30
modelo lógico
adaptações do modelo lógico derivado, 158
conceitos da derivação de modelos relacionais, 159
derivação, 157
restrições, 255
domínio, 255
características conceituais, 256
condicionalidade, 256
definições de tipos de dados especiais, 257
especificação de intervalos, 257
especificação explícita de conjuntos de valores, 257
representação interna, 256
tipo de dados, 256
implementação, 262
direta ou manuseio de estruturas, 262
itens de repetição, 262
integridade, 258
exclusão de ocorrências, 260
manutenção de sincronismo, 261
referências a tabelas inexistentes, 260
referências de chaves estrangeiras a valores de chaves primárias, 259
restrição de chaves estrangeiras, 259
tuplas em modo cascata do tipo "hierarquia", 261
unicidade de valores, 258
Modelo Lógico de Dados (MLD), 29
modelos
conceitos
chave candidata e chave primária (PK), 162
chave estrangeira (FK), 166
domínios de atributos, 167
nulos desconhecidos, 170
nulos inaplicáveis, 169
de dados corporativos, 265, 270
definição, 7
evolução da semântica com passar do tempo, 3
implementação, 161
relacionais, 161

normalização, 173
após o processo de derivação do modelo lógico, 176
benefícios, 176
economia, 181
estabilidade, 176
impactos sobre a estabilidade lógica das aplicações, 177
impactos sobre a estabilidade lógica dos dados, 178
fidelidade ao ambiente observado, 182
flexibilidade, 179
integridade, 179
durante a derivação do modelo lógico, 175
durante o processo de concepção do modelo conceitual, 174
primeira forma normal (1FN), 185
anomalias em tabelas, 198
atualização, 199
eliminação, 199
inserção, 199
definição, 185
exemplo, 191
itens de repetição, 193
processo de obtenção, 188
atributos compostos e atributos atômicos, 189, 194
chaves candidatas e chave primária, 189
grupos e itens repetitivos, 190
segunda forma normal (2NF)
análise do valor parcial ou total de colunas selecionadas, 203
anomalias em tabelas, 208
atualização, 208
colunas parcialmente dependentes, 204
colunas que não participam da chave primária, 203
definição, 200
eliminação, 208
inserção, 208
processo de obtenção, 203
terceira forma normal (3NF), 209
anomalias em tabelas, 217
atualização, 217
colunas transitivamente dependentes, 212
definição, 209
eliminação, 217
inserção, 217
modelo obtido, 218
processo de obtenção, 211
unicidade nas linhas de tabelas, 195
notações gráficas da modelagem conceitual, 138
Bachman, 143
IDEF1X, 138
atributos, 142
entidades, 139
relacionamentos, 140
subtipos e supertipos, 143
"Pé-de-galinha" (*crow foot*), 146
atributos, 147
entidades, 147
relacionamentos, 147
Peter Chen
atributos, 152
entidades, 151
estrutura de agregação, 155
estrutura de especialização-generalização, 154
relacionamentos, 153
Setas, 145

Object-Oriented Modeling and Design (Rumbaugh), 2, 5
Object-Oriented Systems Analysis, Modelling the World in Data (Shlaer & Mellor), 1, 5, 37
objetos
enquadramento em classes e entidades, 3
modelados
dicionarização, 46
observados, 8
captura de aspectos de relacionamento entre, 23
representação de ambientes, 19

relacionamentos
1:N com atributos, 227
1:N sem atributos, 226
caracterização, 75-76
denominação, 72-75
derivação, 220
1:1 com atributos, 223
1:1 sem atributos, 220
dicionarização, 106
existência simultânea, 94
contingentes, 96
independentes, 94
mutuamente exclusivos, 97
grau ou cardinalidade, 76-77

identificação dos relacionamentos, 66-75
relacionamentos entre objetos de tipos diferentes, 67-70
relacionamentos entre objetos do mesmo tipo, 70-72
M:N com ou sem atributos, 230
mudanças na notação de relacionamentos binários e ternários, 88
representação segundo James Martin, 88
representação segundo Peter Chen, 88
números de elementos que participam de um relacionamento, 85-87
presença de atributos, 99
com atributos, 100
mudanças na representação em função de relacionamentos com atributos, 105
sem atributos, 104
presença dos elementos, 89
condicionais, 90
incondicionais, 90
processo de definição e notação, 77
conceituação de relacionamentos 1:1, 78
conceituação de relacionamentos 1:N, 80-82
conceituação de relacionamentos M:N, 82
estabelecimentos de graus de relacionamentos, 78
mudanças na notação de relacionamentos M:N, 84

schema, níveis de, 25-26

Técnicas Estruturadas e CASE (James Martin), 156
"The Entity-Relationship Model: Toward The Unified View of Data" (Peter P. Chen), 33

UNIX, 30

verificação e validação, favorecimento do processo de, 22
visões (*views*), 25
VMS, 30

Este livro foi impresso nas oficinas gráficas da
Editora Vozes Ltda.,
Rua Frei Luís, 100 — Petrópolis, RJ,
com filmes e papel fornecidos pelo editor.