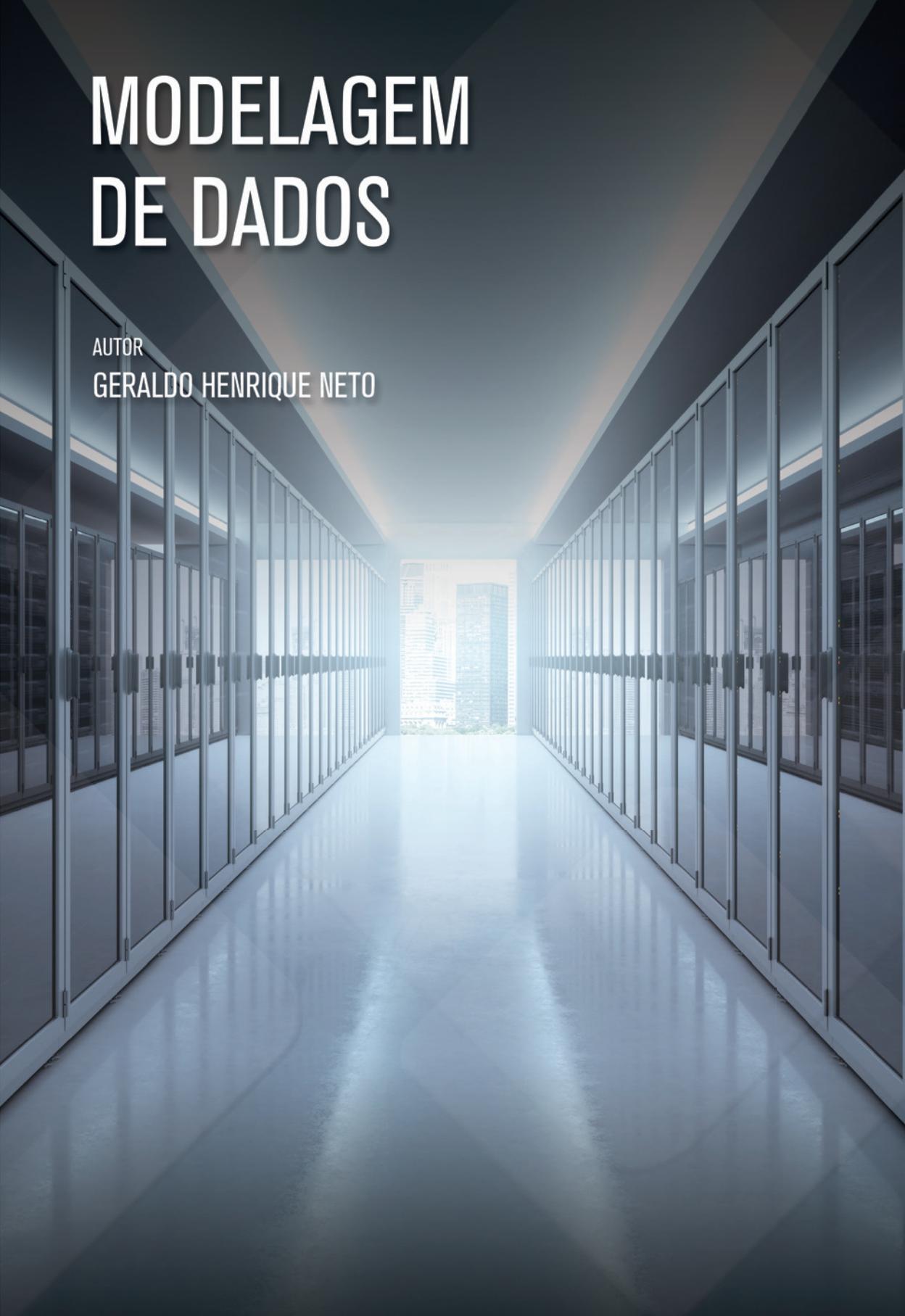


MODELAGEM DE DADOS

AUTOR

GERALDO HENRIQUE NETO



MODELAGEM DE DADOS

AUTOR DO ORIGINAL

GERALDO HENRIQUE NETO

1^a EDIÇÃO

SESES

RIO DE JANEIRO 2015



Estácio

Conselho editorial FERNANDO FUKUDA, SIMONE MARKENSON, JEFERSON FERREIRA FAGUNDES

Autor do original GERALDO HENRIQUE NETO

Projeto editorial ROBERTO PAES

Coordenação de produção RODRIGO AZEVEDO DE OLIVEIRA

Projeto gráfico PAULO VITOR BASTOS

Diagramação FABRICO

Revisão linguística ADERBAL TORRES BEZERRA

Imagen de capa NOME DO AUTOR — SHUTTERSTOCK

Todos os direitos reservados. Nenhuma parte desta obra pode ser reproduzida ou transmitida por quaisquer meios (eletrônico ou mecânico, incluindo fotocópia e gravação) ou arquivada em qualquer sistema ou banco de dados sem permissão escrita da Editora. Copyright SESES, 2015.



Diretoria de Ensino — Fábrica de Conhecimento
Rua do Bispo, 83, bloco F, Campus João Uchôa
Rio Comprido — Rio de Janeiro — RJ — CEP 20261-063

Sumário

Prefácio	7
1. Visão Geral: Banco de Dados e os Sistemas de Gerenciamento de Banco de Dados (SGBD)	10
Visão geral: Banco de dados	11
Dados versus informação	12
Classificando os bancos de dados	13
Sistemas de Arquivos	16
SGBD (Sistema de Gerenciamento de Banco de Dados)	19
Os Usuários de Banco de Dados	22
O SGBD e suas Funcionalidades	24
Vantagens do SGBD	25
2. Projeto de Banco de Dados	32
Projeto de banco de dados	33
Modelo externo	34
Modelo Conceitual	36
Modelo interno	37
Modelo físico	38
Modelo de dados	39

3. Modelo Entidade-Relacionamento	52
Modelo Entidade-Relacionamento	53
As Principais Características do MER	57
Modelo Entidade-Relacionamento Estendido	65
Diagrama Entidade-Relacionamento (DER)	74
Modelando o “negócio”	80
4. Modelo de Dados Relacional	86
Modelo De Dados Relacional	87
Chave Primária (Atributo Identificador)	92
Restrições de Integridade	94
Mapeamento do MER para o Modelo Relacional	102
5. Normalização	122
Normalização	123

Prefácio

Prezados(as) alunos (as)

A cada minuto, uma quantidade significativa de dados serão gerados, independentemente do cenário, seja ele empresarial ou não. Com o surgimento das mídias sociais a geração de informações cresceu de maneira avassaladora, provenientes de diversas fontes distintas, a citar, smartphones, redes de relacionamentos, dentre outros.

Para atender adequadamente essa demanda, às tecnologias de banco de dados, sobretudo o Big Data, o qual é considerado um tema emergente que discorre sobre a evolução das tecnologias de datawarehouse e business intelligence, cujo objetivo é proporcionar o armazenamento e análise de dados, sejam esses, estruturados, semiestruturados e ou não estruturados considerando grandes volumes de dados.

Dessa maneira, considerando a importância em se realizar um armazenamento adequado desses dados para a obtenção do êxito empresarial, independentemente da regra de negócio, a disciplina Modelagem de Dados tem como propósito explorar os principais assuntos acerca de banco de dados.

Para tanto, o conteúdo abordado na disciplina está segmentado da seguinte forma:

Capítulo 1 – estudaremos sobre o surgimento e a evolução dos bancos de dados, apresentando também as principais características dos Sistemas Gerenciadores de Bancos de Dados (SGBD).

Capítulo 2 – abordaremos as fases constituintes de um projeto de banco de dados considerado bem estruturado.

Capítulo 3 – esmiuçaremos o Modelo Entidade-Relacionamento detalhando seus principais componentes.

Capítulo 4 – veremos os principais conceitos do Modelo de Dados Relacional.

Capítulo 5 – abordaremos as principais regras aplicadas ao processo de normalização de dados.

Bons estudos

Profo. Me. Geraldo Henrique Neto

1

Visão Geral: Banco de Dados e os Sistemas de Gerenciamento de Banco de Dados (SGBD)

1 Visão Geral: Banco de Dados e os Sistemas de Gerenciamento de Banco de Dados (SGBD)

Previamente, antes de iniciarmos o estudo acerca da Modelagem de Dados, devemos constituir uma base sólida de conhecimentos triviais que serão imprescindíveis para elucidar os conteúdos futuros, permitindo, dessa forma, uma melhor abstração de outros conceitos gerais no que se refere à Modelagem de Dados.

Acreditamos que você desperte sua curiosidade e se prepare para começar os estudos desta disciplina, estes, sem dúvida, serão de grande relevância para sua formação profissional.

Você se encontra preparado para conhecer os objetivos deste capítulo?!



OBJETIVOS

Nesse primeiro capítulo, exploramos os aspectos históricos relacionados aos bancos de dados, sobretudo, sobre a sua evolução desde o início do seu surgimento até a atualidade. Conhecer os Sistemas Gerenciadores de Banco de Dados (SGBD), realizando uma breve comparação sobre os SGBD e Sistemas de Gerenciamento de Arquivos. Estudaremos também os principais usuários no nível de banco de dados e, por fim, estudaremos as particularidades do SGBD e suas principais funcionalidades.

Exploraremos, de maneira superficial, aspectos históricos, para que você comprehenda como ocorreu a evolução dos Sistemas de Gerenciamento de Banco de Dados, bem como, sua necessidade, sendo assim, estudaremos as primeiras maneiras de administração de dados, por meio de arquivos e sistemas de arquivos, apresentando suas vantagens e desvantagens.



REFLEXÃO

Você já deve ter ouvido falar muito em bancos de dados em diversos momentos de sua vida. Mas o que são bancos de dados? De forma geral, arquivo em que se armazene dados pode de alguma forma entrar nessa categoria. No entanto, de forma específica, há formas eficientes de se armazenar, gerenciar e acessar através de softwares e formatos de dados específicos. Se você nunca lidou com bancos de dados, deve estar se perguntando: "Por onde começo?".

1.1 Visão geral: Banco de dados

Será que você consegue identificar quais foram os motivos que levaram os a realização de investimentos maciços nos estudos destinados ao armazenamento de dados em meios mais acessíveis e eficazes?

Há tempos atrás, organizações empresariais de pequeno, médio e grande porte gastavam cifras significantes de recursos financeiros na contratação de um número expressivo de pessoas para trabalhar no processo de armazenamento e organização de vários arquivos, à mencionar, arquivos referente as informações de clientes, produtos, funcionários, produção industrial, etc.

Em meados da década de 70, a empresa intitulada de IBM, juntamente com seu colaborador chamado de Ted Codd, considerado pioneiro na publicação de um artigo científico, o qual discorria sobre banco de dados relacionais, sugeriu o uso de cálculo e álgebra relacional. Esses dois temas permitia aos usuários finais a manipulação de informações.

O objetivo principal de Codd era implementar um sistema onde fosse possível o usuário manipular as informações, essas armazenadas em tabelas (relação), esse o motivo do termo relacional, por meio do uso de instruções específicas.

Diversas pessoas consideraram não adequadas aplicar as notações matemáticas de Codd na manipulação de informações provenientes de tabelas, não despertando assim, o interesse da maioria das pessoas. Entretanto, ninguém imaginava que as teorias propostas por Codd se tornaria conceitos triviais na evolução no que tange o armazenamento e recuperação de informações.

Nesse mesmo período, a IBM criou o *System R*, o qual promoveu subsídios para a elaboração de um protótipo de banco de dados relacional, conduzindo mais adiante, a criação da linguagem SQL (*Structured Query Language*) e do conhecido DB2. Não restam dúvidas que, na área de banco de dados, a linguagem SQL é considerada um padrão para os principais fabricantes de bancos de dados relacional.

Inoportunamente, a IBM deixou o *System R* em segundo plano por um grande período (diversos anos). O interesse da IBM era em um sistema de banco de dados com credibilidade que utilizava tecnologia de ponta, conduzindo, no ano de 1968, o IMS. Assim, a IBM não mediou esforços em publicar os resultados dos trabalhos científicos produzidos pela sua equipe, ora responsável pelo *System R*. Posteriormente, o criador de uma pequena e singela empresa, Larry Ellison, se deparou com o artigo publicado pela equipe da IBM, e, imediatamente, deu início ao processo de contratação dos desenvolvedores do *System R* e da Univer-

sidade da Califórnia. No ano de 1979, graças ao perfil empreendedor de Larry, um novo banco de dados relacional com base na linguagem SQL foi lançado no mercado, bem antes da conservadora IBM.

Uma versão de banco de dados intitulada de Oracle, em 1983, foi criada pela empresa de Ellison, incrementando o faturamento anual em 5 milhões de dólares.

1.2 Dados versus informação

É normal a utilização da denominação informação, quando na verdade estamos nos referenciando a um dado, ou até mesmo, um conjunto de dados. Na prática, nas empresas essa substituição se tornou relativamente normal. Todavia, é imprescindível destacar que dado e informação, embora intimamente ligados, possuem conceitos distintos. Saber diferenciar os é importante, sobretudo para a análise da qualidade da informação.

Imagine que você seja responsável por identificar a satisfação de uma parcela de usuários no que diz respeito a um sistema computacional especialista. Você poderia inicialmente, obter essa informação realizando entrevistas aos usuários do sistema, questionando os mesmos sobre determinados aspectos pontuais.

Pensando em facilitar seu trabalho, você poderia elaborar um formulário on-line, disponibilizado na Web, permitindo a disseminação do mesmo entre os diversos usuários. Posteriormente, ao armazenamento dos dados das inúmeras respostas provenientes do formulário on-line, temos os dados em sua forma bruta, isto é, eles não possuem nenhum significado. Dessa maneira, necessitamos transformá-los em dados que possam gerar algum tipo de informação. A lapidação dos dados brutos, torna o mesmo com maior valor agregado, possibilitando extraírmos respostas objetivas e eficazes do repositório, como exemplo, podemos mencionar: “O módulo responsável para geração dos relatórios analíticos são de fácil usabilidade?”

Sendo assim, podemos contextualizar que quando os dados brutos são manipulados adequadamente, podemos extrair o seu significado e, dessa forma, gerar as informações necessárias para uma possível tomada de decisão.

Para alguns, esse tipo de processo de gerar informação a partir de dados brutos, pode ser considerado simples, ou complexo, principalmente se considerarmos a confecção de relatórios estatísticos.

Esse outro exemplo, refere-se a utilizar um dado que ora representa uma temperatura qualquer. Nenhuma novidade de considerarmos 43º como sendo

um dado em sua forma bruta. Porém, nesse exemplo, 43º não está vinculado a nenhuma informação, ou seja, não possui significado algum, ao menos que, seu contexto seja evidenciado, como: Esse valor representa a temperatura da cidade de Ribeirão Preto em plena primavera? Ou, esse valor de temperatura é a média de temperatura trimestral da região metropolitana de São Paulo?

Dessa forma, é correto dizer, que os dados constituem os pilares para que alcancemos as informações, tornando-se quesito extremamente importante na tomada de decisão empresarial, independentemente da área de atuação e porte, sejam empresas governamentais, privadas e ou de prestação de serviço.

Retomando nosso exemplo inicial, os dados provenientes das questões do formulário on-line, podem identificar os principais pontos favoráveis ou não favoráveis do módulo responsável pela geração de relatórios analíticos, conduzindo a uma melhor tomada de decisão a fim de atender os principais apontamentos de seus usuários.

1.3 Classificando os bancos de dados

Podemos classificar os bancos de dados de acordo com suas características e aplicabilidade dos dados neles armazenados. Por exemplo, essa classificação pode variar de acordo com o número de usuários, localização, e, tipo de uso. Na sequência, correlacionaremos os principais tipos de banco de dados. É importante mencionar que, em algumas literaturas, essa classificação pode possuir discretas alterações.

A primeira classificação é baseada pelo número de usuários:

- Banco de Dados Monousuário: esse tipo de banco de dados promove o suporte de apenas um único usuário por vez. Ou seja, para que você possa entender, imagine que o usuário Fernando Collor se encontra utilizando o banco de dados, e os usuários Fernando Henrique e Paulo Maluf, por sua vez, necessitam aguardar o usuário Fernando Collor finalizar sua transação para obter o respectivo acesso ao mesmo repositório (banco de dados). Nesse tipo de banco de dados, não existe, em hipótese alguma, o que denominamos de controle de concorrência;
- Banco de Dados Multiusuário: ao contrário do banco de dados monousuário, esse tipo de banco de dados dá suporte e gerência diversos usuários simultaneamente.

Na sequência, apresentaremos outro tipo de classificação, essa considera a localização do banco de dados:

- Banco de Dados Centralizado: realiza a centralização dos dados em um único local fisicamente existente, realizando o devido suporte ao mesmo;
- Banco de Dados Distribuído: nesse tipo de banco de dados o suporte é destinado a distribuição dos dados por diversos locais, esses, normalmente, residente em locais fisicamente distintos.

Uma das mais popular classificação de banco de dados refere-se em como os dados são manipulados, respeitando sua evolução no tempo, ou seja, gerencian-do dados on-line e históricos. Essa classificação pode ser determinada como:

- Banco de Dados Operacional: realiza o suporte a transações empresariais diárias. Esse tipo de banco de dados também pode ser classificado como Banco de Dados Transacional ou, até mesmo, de Banco de Dados de Produção;
- Data Warehouse: esse tipo de banco de dados, a ênfase é no armazena-mento de dados que promovem o suporte na extração de informações valiosas no que se refere a tomada de decisão. A fim de conduzir adequa-damente essas decisões, os dados carecem de tratamento específico, ou seja, torna-se necessário realizar tratamento, cujo objetivo é a obtenção de informações valiosas, a citar, avaliação semestral de venda de uma de-terminada região, dentre outros. Um *Data Warehouse* possui a estrutura bem diferente do Banco de Dados Operacional, por tornar ágil o acesso a esses dados;
- Banco de Dados Temporal: esse tipo de banco de dados permite que os usuários manipulem os dados de acordo com suas necessidades, permitindo assim o armazenamento de todo o histórico das modificações aplicadas aos dados. Para exemplificar, considere aplicações que necessitam de promo-ver o controle de acesso aos dados temporais, como: sistemas médicos (evo-lução dos tratamentos aplicados aos pacientes), sistemas empresariais (sis-temas financeiros, monitoramento da produtividade, gestão de RH, etc.), sistemas de informações geográficas (SIG), dentre outros;
- Banco de Dados Espacial: permitem recuperar objetos considerando um espaço multidimensional. Um caso especial desse tipo de banco de dados é o banco de dados geográfico, por garantir o armazenamento de dados referente a localização de rios, municípios, estados, rodovias, etc.;

- Banco de Dados Meteorológico: pela sua própria nomenclatura, refere-se a um tipo de banco de dados que armazena dados sobre o tempo (temperatura, umidade do ar, velocidade do vento, etc.);
- Banco de Dados Multimídia: são normalmente direcionados ao armazenamento e manipulação de dados considerados não convencionais, como, imagens, vídeos, músicas, etc.;
- Banco de Dados Especialista: você poderá encontrar, em algumas literaturas, um outro nome aplicado a esse tipo de banco de dados (banco de dados/sistemas baseados em conhecimento). Esse tipo de banco mistura raciocínio e inferência, por meio do uso de inteligência artificial.

É ainda possível classificarmos os bancos de dados levando em consideração como os dados estão estruturados. Um dado não estruturado é considerado como sendo um dado em sua forma original, isto é, em sua forma bruta, sem nenhum tipo de lapidação/tratamento. Para tanto, esse tipo de dado não permite que realizemos o devido processamento para alcançarmos a valiosa informação agregada nele. Ao contrário, dado estruturado pode ser considerado como o resultado de um processo que utilizou dados não estruturados, atendendo a uma pré-formatação, criando facilitadores no que se refere ao armazenamento, a manipulação e obtenção de informação.

Em ambientes empresariais, certamente você irá se deparar com o armazenamento e manipulação de dados semiestruturados, ou seja, um tipo de dado parcialmente processado. Atualmente, existe uma demanda considerável para armazenar e gerenciar dados não estruturados e semi-estruturados, conduzindo ao surgimento de mais uma nova vertente de banco de dados, os intitulados de Banco de Dados XML (*Extensible Markup Language*).

Para encerrar esse tópico, devemos deixar explícito que os termos banco de dados e base de dados não podem ser considerados como sinônimos, isso porque, uma base de dados, normalmente, é gerenciado por ferramentas de suporte a decisão empresarial, como exemplo, podemos mencionar, o sistema ERP (Enterprise Resource Planning). Entretanto, o gerenciamento de um banco de dados ou um conjunto de banco de dados é promovido pelo uso de um sistema de gerenciamento de banco de dados (SGBD).

1.4 Sistemas de Arquivos

Você já se deparou com as siglas FAT, FAT32, NTFS? Algo familiar para você? Bem, pois fique sabendo que essas siglas têm um propósito em comum, e, será que você saberia me responder?

Os sistemas de arquivos, ora relacionados acima por meio das siglas FAT, FAT32, NTFS, por exemplo, possuem como objetivo comum realizar a organização e armazenamento de dados em sistemas de informação computacionais, formando dessa maneira, os dois principais sistemas para gerenciamento e controle de informação, a citar, o próprio sistema de arquivos e o sistema de gerenciamento de banco de dados (SGBD).

Sistemas de arquivos é o mecanismo utilizado para organizar e armazenar dados por meio de uma estrutura lógica, essa responsável pela alocação física dos arquivos em dispositivos de armazenamento (disco rígido, memória flash, CD-ROM, DVD-ROM, dentre outros.). Bem, para que você possa abstrair ainda mais esse conceito, considere um sistema de arquivos como sendo uma estrutura responsável por delimitar como os arquivos serão efetivamente gravados e armazenados em qualquer dispositivo de armazenamento.

Um detalhe também considerado importante para o conceito ora exposto, é que todo esse controle destinado ao acesso aos dispositivos de armazenamento é promovido pelos sistemas operacionais, a citar, MS-DOS, Windows, Linux e Unix. Basicamente, a maioria dos sistemas de arquivos faz uso dos recursos de armazenamento de dados os quais utilizam vários blocos nomeados de setores, esses com tamanhos já pré-definidos.

Não temos dúvidas que você, eventualmente, utiliza um desses sistemas de arquivos FAT (File Allocation Table), FAT32 e NTFS (New Technology File System), da família dos sistemas operacionais da Microsoft. E daí? Você quer saber qual é realmente a diferença existente entre esses sistemas de arquivos? Nossa objetivo não é esmiuçar esses detalhes, e sim, promover uma visão resumida, isto é, a principal diferença entre esses sistemas de arquivos está relacionada às limitações da tecnologia que por sua vez, foram evoluindo com o transcorrer do tempo, bem como a inevitável melhoria da segurança e capacidade de armazenamento.

Bem, agora imagine que você tenha a necessidade de gravar um vídeo (filme de DVD) em uma mídia de armazenamento específica, para exemplificar, o seu próprio *Pen-Drive*. Você tem ideia de como o sistema de arquivo irá trabalhar? Antes de qualquer coisa, é necessário conhecer o funcionamento de um sistema de arquivo, que utiliza um tipo de tabela para armazenar todos os detalhes da localização dos dados de cada arquivo. Por exemplo, o sistema de arquivo FAT fragmenta a área do dispositivo de armazenamento em pequenos blocos, permitindo assim que um único arquivo possa ocupar diversas posições distintas, onde, esses blocos, necessariamente, não necessitam estar alocados continuamente.

Podemos considerar que os sistemas de arquivos foram os pioneiros no que se refere aos sistemas de armazenamento e manipulação de dados. Tais sistemas sofreram melhorias ao longo de décadas, cujo objetivo era acompanhar a evolução tecnológica. Entretanto, infelizmente, alguns problemas podem ser caracterizados, a citar:

- A estrutura de arquivos é definida pelo próprio código-fonte do sistema computacional, prejudicando consideravelmente sua manutenção;
- O controle de acesso desses arquivos apresentam grandes obstáculos quando mencionamos o compartilhamento;
- A falta de compatibilidade entre os sistemas computacionais prejudica consideravelmente a funcionalidade dos sistemas, pelo simples fato de criar arquivos e sistemas voltados a um único e exclusivo sistema operacional, e que, na maioria das vezes são realizados por diversos desenvolvedores distintos que utilizam linguagens de programação díspares;
- Sistemas computacionais com dados distintos conduz a inconsistência dos mesmos;
- O uso de formatos específicos acarreta o isolamento de dados;
- A duplicidade não supervisionada dos dados leva a redundância dos mesmos;
- O fácil acesso aos dados direciona a sérios problemas de segurança;
- A falta de controle de acesso concorrente promovido por diversos usuários acessando o mesmo dado.

A Figura 1.1 abaixo apresenta um exemplo do armazenamento de dados por meio de sistemas de arquivos. Verifique que, toda a segurança e controle de acesso aos dados são de responsabilidade da aplicação.



Figura 1.1 – Esquema Simplificado do Sistema de Arquivos

Uma dúvida que pode ter surgido em você, diz respeito se a utilização de sistemas de arquivos no gerenciamento e manipulação dos dados é uma boa alternativa, nos dias de hoje? Para que você possa contextualizar ainda mais os problemas provenientes do uso de sistemas de arquivos, principalmente, o que se refere à inconsistência e controle de acesso, suponha que dois clientes estejam buscando produtos ora disponíveis para comercialização em um e-commerce qualquer. Ainda para agravar a situação, o e-commerce possui apenas uma única unidade de um DVD, e, prontamente, o sistema apresenta essa informação para ambos os clientes. Ao mesmo tempo, os dois clientes escolhem o mesmo DVD para aquisição. Percebe-se que essa simples transação acarretará em problema no sistema, promovendo a inconsistência dos dados. Esse problema que gerou a inconsistência foi produzido devido à ausência de bloqueio da compra de um dos clientes, e a exibição de uma mensagem informando que não existe mais o produto em estoque.

Dessa maneira, percebe-se que o armazenamento e manipulação de dados por meio do uso de sistemas de arquivos podem levar a diversos dissabores. Esse mesmo exemplo hipotético pode ser ampliado para outros tipos de sistemas, tais como, sistemas bancários, sistemas de hotelaria, sistemas hospitalares, etc..

Devido a alguns problemas apresentados pelo uso de sistemas de arquivos para o armazenamento de dados, e, pela necessidade de utilizar novas estruturas de armazenamento nesses sistemas computacionais, como também, o alto custo aplicado na manutenção para solucionar esses dissabores, criou-se a evolução dos sistemas de arquivos, os extraordinários sistemas de gerenciamento de banco de dados (SGBD), que utilizam novas estruturas para o armazenamen-

to e gerenciamento dos dados.

A seguir, iniciaremos um estudo detalhado acerca dos sistemas de gerenciamento de banco de dados, apresentando suas principais vantagens em relação aos sistemas de arquivos.

1.5 SGBD (Sistema de Gerenciamento de Banco de Dados)

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é constituído por um conjunto de aplicativos que possibilitam a criação e manipulação de diversos bancos de dados. Como exemplo de SGBDs mais utilizados no mercado, podemos citar o *MySQL*, *Oracle*, *FireBird*, *DB2*, *SQL Server* e *PostgreSQL*.

Quando se utiliza um SGBD, o acesso e o gerenciamento dos dados são promovidos pelo próprio SGBD, diferenciando-se do armazenamento e gerenciamento de dados por meio de sistemas de arquivos. Ou seja, se assemelha a uma interface inserida entre o banco de dados físico (discos, mídias de armazenamento, etc.) e os usuários. Como você pode visualizar na Figura 1.2, o SGBD recebe todas as requisições de inúmeras aplicações computacionais, realiza a devida interpretação a fim de melhor atende-las, escondendo a complexidade interna existente dos bancos de dados dos usuários finais.

CONEXÃO

SGBD é uma coleção de aplicativos computacionais responsáveis pelo gerenciamento de uma conjunto de banco de dados. (Rob et al., 2011).

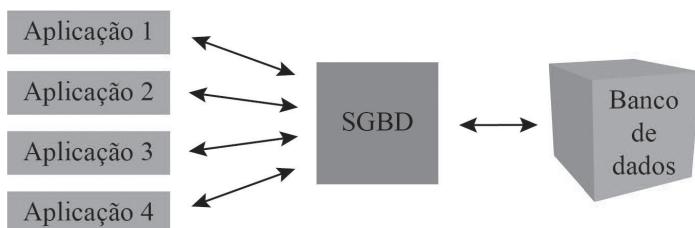


Figura 1.2 – Esquema Simplificado do Sistema Gerenciador de Bancos de Dados

Tais aplicativos, normalmente, são desenvolvidos por uma equipe de desenvolvedores através da utilização de uma linguagem de programação específica, como podemos mencionar: Java, PHP, C/C++, Pascal, C#, etc. ou, esporadicamente, constituídos a partir do uso de módulos proprietários de um SGBD, como exemplo, podemos evidenciar o Forms e Reports da própria *Oracle*.

Você já deve ter percebido que os dados são caracterizados como sendo a matéria bruta e imprescindível para que alcancemos qualquer tipo de informação, impulsionando ainda mais a necessidade de utilizarmos métodos eficientes e eficazes para manipula-los.

Algumas vantagens na utilização de um sistema de gerenciamento de banco de dados associados aos aplicativos computacionais podem ser evidenciados, a citar, o SGBD, diferentemente do sistema de arquivo, possibilita, de maneira facilitadora, o compartilhamento dos dados para diversos sistemas e usuários; também permite a integração das várias visões de dados de cada usuário em um único repositório de dados; fornece também modelos capazes de incrementar de maneira significativa a privacidade e segurança dos dados e por fim, auxilia na redução de eventuais inconsistências dos dados.

Informações de qualidade são, normalmente, oriundas de um processo de manipulação e gerenciamento de dados bem sucedidos, que por sua vez, criam subsídios para uma melhor tarefa de tomada de decisão empresarial. Essas vantagens elucidadas acima, no que tange a utilização de um SGBD não se limitam a apenas esses fatores. Não temos dúvida que ao longo do processo de aprendizagem, como também, de sua via profissional, você irá se deparar com diversas outras vantagens ao esmiuçar os detalhes dos bancos de dados, sobretudo, na escolha de um projeto adequado na aquisição do mesmo.



CONEXÃO

Leia sobre a relação: Dados-Informação-Conhecimento em: <http://www.ime.usp.br/~vwsetzer/dado-info-Folha.html>

Este outro artigo trata sobre Inteligência Competitiva das Organizações, pautando-se na relação que estudamos:

Os usuários dos SGBDs sejam eles, usuários finais, programadores e administradores de banco de dados, em sua rotina diária, necessitam de extrair conhecimento e pleno controle dos recursos disponíveis para usufruir totalmente de todos os seus benefícios, tais como, podemos evidenciar:

- Maior velocidade em produzir acesso aos dados;
- Solucionar problemas provenientes da ausência de integridade e redundância de dados;
- Simplicidade na criação e gerenciamento de bancos de dados;
- Certificar-se se os dados estão disponíveis para a distribuição física.

Infelizmente, a ausência de conhecimentos triviais dos recursos disponíveis dos SGBDs pode promover uma péssima manipulação dos dados e, inevitavelmente, provocar falhas no acesso, produzindo assim, o que denominamos de falha de segurança.

Após o surgimento dos SGBDs, todo aquele trabalho desempenhado pelos programadores para realizar o devido controle de acesso aos dados, implementar a integridade e evitar redundância dos dados, felizmente, deixaram de ser necessários, entretanto, se considerarmos alguns itens, os sistemas de arquivos podem possuir benefícios superiores aos SGBDs. Não tenha dúvida que essa afirmação lhe parece muito estranha, mas por mais bizarro que ela possa ser, a performance, quando utilizamos sistemas de arquivos, é considerada superior em relação ao dos SGBDs, isso porque é realizado a implementação específica e exclusiva destinada a uma determinada aplicação, já os SGBDs são utilizados por aplicações consideradas genéricas, visto que, o objetivo é atender diversas e variadas necessidades.



ATENÇÃO

Vimos que dado e informação não são sinônimos, e que a informação é resultante de um processamento aplicado no dado bruto, considerando um determinado contexto, ou seja, a informação é o dado atribuído a um determinado significado, assim temos:

$$\text{Informação} = (\text{dado} + \text{significado}).$$

1.6 Os Usuários de Banco de Dados

Nesse tópico, evidenciaremos os principais tipos de usuário que lidam com banco de dados. Na maioria das literaturas existentes acerca de banco de dados, podemos encontrar quatro tipos distintos de usuários de banco de dados, os quais são diferenciados exclusivamente pela forma como interagem com o sistema de banco de dados. Existem também, associado a esse conceito, vários tipos de interfaces, essas normalmente, projetadas para atender os diversos tipos de usuários. A seguir, esmiuçaremos os quatro tipos de usuários, dando exemplos de suas principais atribuições:

- Usuários finais: esse tipo de usuário não possui conhecimento avançado, e, normalmente estabelecem interação com o banco de dados por meio do uso de aplicações computacionais previamente implementada. Como exemplo, imagine um caixa bancário que necessita realizar a transferência de R\$ 100,00 de uma conta corrente X para outra conta corrente Y. Nesse caso, o usuário final, o caixa da instituição financeira, necessitará invocar o aplicativo para que seja possível realizar a transferência do valor. Por outro lado, esse aplicativo solicita ao usuário final (o caixa da instituição) o valor e a conta para a qual o dinheiro será transferido. Tomamos como outro exemplo, um usuário final que deseja consultar seu saldo bancário por meio do uso de um aplicativo *Web (Home Bank)*. Esse usuário deverá acessar um formulário, no qual será necessário a inserção do número de sua agência, conta corrente e senha, para identificação do mesmo. Um aplicativo residente no servidor Web, por sua vez, recupera o saldo da conta corrente, utilizando o número da agência e conta corrente informado pelo usuário final, e apresentando essa informação para o mesmo. Normalmente, a interface utilizada pelos usuários finais é constituída por formulários, os quais o usuário pode inserir as variáveis necessárias que contemple o formulário. Também é possível que usuários finais simplesmente leia relatórios produzido pelo banco de dados;
- Desenvolvedores de Aplicativos: tipo de usuário caracterizado por desenvolver aplicativos computacionais. Os desenvolvedores de aplicativos selecionam, entre diversas ferramentas disponíveis, uma que mais lhe adeque a fim de promover o auxílio no desenvolvimento de interfaces com o usuário. Como exemplo, podemos correlacionar a ferramenta RAD (*Rapid Application Development*), a qual permite que um determinado desenvolvedor de aplicativos crie formulários e relatórios sem muito esforço de programação;

- **Usuários Avançados:** são denominados de usuários avançados aqueles que escrevem aplicações voltadas a banco de dados especializados, ou seja, aplicativos que normalmente não se acopla a estrutura de processamento de dados convencional. Como exemplo desses aplicativos, podemos citar: sistemas de base de conhecimentos, sistemas especialistas, sistemas que armazenam e manipulam dados com tipos de dados considerados complexos (som, vídeo e imagem) e sistemas de modelagem de ambiente.

Acredito que você deve ter reparado que havíamos comentado sobre quatro tipos principais de usuários de banco de dados, e, correlacionamos acima apenas três tipos deles. Não se preocupe, o último usuário, intitulado de Administrador de Banco de Dados merece uma ênfase maior, se compararmos com os demais.

Um dos objetivos principais de utilizarmos um SGBD é obtermos controle total sobre os dados e os aplicativos que promovem o acesso aos mesmos. Um usuário responsável por promover esse controle sobre o SGBD é chamado de administrador de banco de dados (DBA). A seguir, apresentaremos suas principais atribuições:

- Por meio do uso de um conjunto de instruções de definição de dados (DDL), o DBA é capaz de criar um esquema de banco de dados qualquer;
- Estabelecer a estrutura de armazenamento, bem como definir mecanismos de acesso aos dados;
- Para melhorar o desempenho do banco de dados, o DBA poderá realizar mudanças no esquema e na organização física, como também, para atender eventuais necessidades específicas da empresa;
- O administrador de banco de dados promove o controle de objetos ora armazenados no banco de dados que os diversos usuários podem acessar, concedendo distintos tipos de autorização. Essas autorizações, normalmente são armazenadas e mantidas em uma estrutura especial a qual o SGBD consulta sempre que algum usuário tenta acessar os dados;
- Considerada como fundamentais, as atribuições de manutenção do banco de dados, desempenhada pelo DBA são cruciais para promover a continuidade do sistema de banco de dados, a citar: realização de cópias de segurança (backups) periodicamente, independentemente dos dispositivos de armazenamento, ou até mesmo em servidores secundários, a fim de prevenir perda de dados em caso de anomalias, catástrofes, etc. Certificar a existência de espaço suficiente em disco para desempenhar operações convencionais e, caso seja necessário, incrementar o espaço em disco. Para finalizar, o DBA também deverá monitorar tarefas processadas no banco de dados e

garantir que o desempenho não seja afetado por outras tarefas sobrecarregadas emitidas por alguns usuários.

1.7 O SGBD e suas Funcionalidades

Antigamente, as primeiras arquiteturas de SGBD utilizavam computadores de grande porte, denominados mainframes. A utilização do mainframe tornava-se necessário para realizar processamento de todas as funções do sistema, isso incluía os aplicativos, as interfaces com o usuário, bem como todos os demais processos responsáveis por prover as funcionalidades básicas dos SGBDs. Grande parte dos processamentos eram desempenhados de maneira remota, isso é, somente os dados a serem visualizados eram submetidos para os terminais de visualização, ora conectados ao mainframe por meio de uma rede de comunicação.

Atualmente, identificamos o decremento do custo de aquisição de equipamentos de hardware, porém, na contra mão, o aumento significante da capacidade de processamento e armazenamento é digna de nota. Isso favoreceu imensamente a substituição dos terminais por computadores pessoais pelos usuários. No início, os SGBDs faziam uso dos computadores pessoais da mesma forma que utilizavam os terminais, isto é, o SGBD trabalhava de maneira centralizada e toda as demais funcionalidades, como, a execução de programas aplicativos e processamento de interface do usuário eram processados em uma única e exclusiva máquina.

Com o transcorrer do tempo os SGBDs começaram a usufruir do processamento existente do lado do usuário, que por sua vez, assumia o papel de “cliente” do SGBD. Essa filosofia conduziu a uma arquitetura denominada cliente/servidor, que se destina a segmentar os processamentos provenientes dos aplicativos, esses localizados do lado do cliente e o gerenciamento dos dados, esse sendo executado do lado do servidor.

Dessa forma, essa arquitetura cliente/servidor foi introduzida aos SGBDs comerciais, promovendo o surgimento de diversas técnicas para a implementação dessa arquitetura. As consultas SQL, juntamente com as funcionalidades transacionais residem do lado do servidor, esse intitulado de servidor de consulta ou servidor de transação. É exatamente dessa maneira que as máquinas clientes visualizam o servidor, como sendo um servidor SQL. Cada máquina cliente é responsável por constituir suas consultas SQL e disponibilizar uma interface com o usuário.

Logo abaixo, podemos resumir algumas das arquiteturas triviais dos SGBDs:

- Centralizada: essa arquitetura tem como característica a existência de um computador com expressivo poder de processamento, que lhe proporciona a centralização dos processos do SGBD, como também, permite funcionar como emulador para “n” aplicativos. Sua principal vantagem é possibilitar que diversos usuários manipulem significativas quantidades de dados e, como sua principal desvantagem, podemos citar, seu alto custo de aquisição, sobretudo por exigir cenário especial para mainframes e soluções centralizadas;
- Computador Pessoal (PC): essa outra arquitetura permite que computadores pessoais trabalhem em sistema stand-alone, isso significa que, esses PCs realizam seus próprios processamentos de maneira autônoma. Bem no início a adoção dessa arquitetura, o processamento era considerado limitado, entretanto, com o transcorrer do tempo, paralelamente com a evolução substancial do hardware, atualmente possuímos PCs com larga escala de processamento. Como vantagem dessa arquitetura, podemos considerar sua simplicidade;
- Cliente-Servidor: já nessa arquitetura, o cliente é intitulado de (front-end), esse responsável por executar tarefas do aplicativo, fornecendo uma interface para usuário, seja por meio de telas ou processamento de entrada e saída. O servidor por sua vez, assume o papel de (back-end), por simplesmente executar as consultas no SGBD e devolve-las como resultado ao cliente. Essa arquitetura é muito popular, se compararmos com as demais, vista até o presente momento. Um dos pontos fortes dessa arquitetura é a segmentação do processamento por meio do uso de dois sistemas, reduzindo consideravelmente o tráfego de dados na rede de comunicação.

1.8 Vantagens do SGBD

Vimos que os SGBDs basicamente funcionam como servidores de dados que por sua vez disponibilizam uma interface de comunicação a qual permite que aplicativos computacionais estabeleçam comunicação com ele. Utilizando-se essa interface é possível que os aplicativos realizem a solicitação de inserção, alteração, recuperação ou organização dos dados armazenados pelo SGBD. Bem, você deve estar se questionando: Será que é correto concluir que o SGBD reúne todos os processos organizacionais de uma empresa?

A resposta correta é sim! As empresas são subsidiadas pelos seus dados e no processamento dos mesmos. Não tenha dúvida que o gerenciamento e tratamento correto desses dados irão permitir relevantes tomadas de decisões. Nesse contexto, a informação gerada pelos dados possui um valor imensurável e, basicamente, todo esse cenário computacional é gerenciado pelo SGBD, dessa forma, também é correto dizermos que o SGBD é o “coração” para a grande maioria das empresas. Para melhor abstrairmos todo esse conceito, podemos visualizar por meio da Figura 1.3, como efetivamente o SGBD funciona:

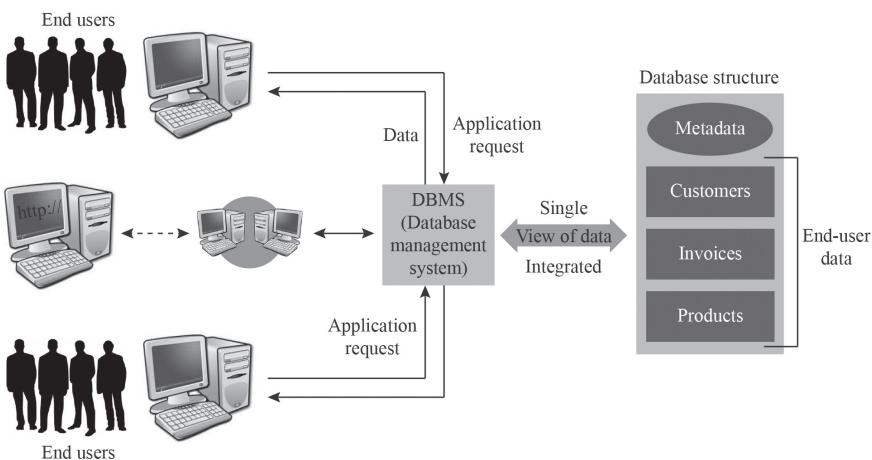


Figura 1.3 – Sistema de Gerenciamento de Banco de Dados. Extraído de: Rob, 2005 p. 8

Podemos visualizar na Figura 1.3 acima que o SGBD é o eixo central que promove acessos dos diversos tipos de usuários e ou aplicativos computacionais aos dados. Isso é o SGBD recebe diversas solicitações dos aplicativos e converte em operações mais elaboradas para o acesso aos dados. Assim, a complexidade de acesso as fontes nativas dos dados são encapsuladas pelo próprio SGBD, reduzindo esforços no que se refere a implementação dessas funcionalidades na aplicação do cliente.

É totalmente indicada a adoção do uso de um SGBD em qualquer cenário empresarial que necessita de manipular dados de maneira correta e em sua totalidade. A seguir apresentaremos as principais vantagens do uso de um SGBD:

- Compartilhamento de Dados: o SGBD fornece mecanismos os quais permitem que os usuários finais consigam acessar os dados facilmente, mesmo lidando com um grande volume de dados;

- Segurança de Dados: em um cenário que possui uma quantidade expressiva de usuários que acessam os dados, os riscos do quesito segurança também são aumentados. Com a adoção dos SGBDs torna-se factível criar um modelo para melhor determinar as políticas de segurança empresarial, promovendo a segurança a nível de usuário, refletindo em uma maior privacidade no acesso aos dados. Diversas regras de segurança podem ser elaboradas determinando assim quais usuários podem ou não acessar o banco de dados e, ainda determinar, quais objetos os mesmos poderão acessar;
- Centralização dos Dados: um benefício importante refere-se a centralização dos dados, sobretudo por permitir que todos os dados possam ser integrados a um único repositório, minimizando dessa forma as redundâncias dos dados;
- Flexibilidade: eventualmente, alterações aplicadas ao banco de dados são necessárias para contemplar os novos requisitos organizacionais. Como exemplo, podemos citar que um determinado usuário carece de uma visão de dados ora não disponível no banco de dados. Grande parte dos SGBDs possibilitam que mudanças estruturais possam ser aplicadas no banco de dados sem ao menos interferir nos aplicativos computacionais existentes;
- Compartilhamento de Dados: a maioria dos SGBDs são multiusuário e precisam promover o controle adequado da concorrência para garantir que as transações simultâneas obtenham êxito ao seu final;
- Múltiplas Interfaces: mediante a existência de diversos tipos de usuários, os quais possuem distintos níveis de conhecimento técnico, um SGBD precisa prover um leque de interfaces para melhor atendê-los. A citar, interfaces para consultas emitidas por eventuais usuários, interfaces para desenvolvedores de aplicativos, interfaces baseadas em formulários destinadas aos usuários finais;
- Gerenciamento e Armazenamento de Dados: o SGBD constitui e administra as estruturas consideradas complexas utilizadas no armazenamento de dados, permitindo que o usuário dê ênfase em suas verdadeiras necessidades empresariais, evitando assim que o mesmo perda o foco em procedimentos complexos de armazenamento de dados em baixo nível;
- Gerenciamento de Transações: uma transação se resume como sendo um conjunto lógico de trabalho. Toda e qualquer transação possui início (BEGIN) e fim (COMMIT e ou ROLLBACK). Dentro desse contexto, todos

os registros relacionados a uma determinada transação são inseridos, alterados ou removidos ou nada é registrado (tudo ou nada), garantindo a consistência e integridade dos dados, ora promovido pelo SGBD.

Para concluir as principais vantagens do uso de um SGBD, todas as transações desempenhadas por ele devem atender as propriedades que denominamos de ACID (Atomicidade, Consistência, Isolamento e Disponibilidade). Essas propriedades melhoram o gerenciamento da concorrência dos dados e objetos junto ao SGBD.



ATENÇÃO

Um sistema de banco de dados é uma coleção de dados inter-relacionados e um conjunto de programas que permitem aos usuários acessar e modificar esses dados (Silberschatz, et al., 2006).



ATIVIDADE

1. Cite pelo menos três exemplos de Dados e Informações, num contexto de empresarial qualquer. Detalhe cada um.
2. Apresente quatro diferenças significativas existentes entre um sistema de arquivo e um SGBD.
3. Identifique cinco características de um sistema de gerenciamento de banco de dados (SGBD).



REFLEXÃO

Atualmente a demanda por informações, independentemente do segmento organizacional é uma constante expressiva. Para você ter ideia, no Facebook, diariamente são publicadas 250.000.000 de fotos, 2.000.000.000 de comentários ou “curtição” de posts, 900.000.000 atrações como comunidades e eventos, 800.000.000 de usuários (só no Brasil, 28.000.000). Vimos que as organizações empresariais manipulam diariamente um imenso volume de dados cujo propósito é torná-los em algum tipo de informação relevante para o suporte a tomada de decisão.

Não se esqueça de que nunca poderemos considerar que dado e informação são sinônimos. Estudamos que anteriormente da existência dos bancos de dados os dados empresariais os mesmos eram manipulados por sistemas de arquivos, de forma manual. É notório que os sistemas de arquivos sobreviveram por um grande espaço de tempo, onde, muitos problemas foram identificados.

Na sequência, estudamos como os sistemas de arquivos e os sistemas de gerenciamento de banco de dados funcionam, detalhando suas principais características e benefícios.

Por fim, aprendemos acerca dos principais tipos de usuários (leigos, desenvolvedores, avançados, especialistas e DBAs), que de alguma maneira, utilizam os recursos de um SGBD.



LEITURA

Artigos on-line: para você aumentar ainda mais o seu nível de aprendizagem envolvendo os conceitos de Dados e Informação e demais assuntos desta unidade, consulte as sugestões de links abaixo:

<http://www.ime.usp.br/~vwsetzer/dado-info.html>

<http://gestorescomvisao.blogspot.com.br/2008/11/dado-x-informao.html>



REFERÊNCIAS BIBLIOGRÁFICAS

ELMASRI, R.; NAVATHE, S.B. Sistemas de bancos de dados. São Paulo: Pearson (Addison Wesley), 2005.

KORTH, H.; SILBERCHATZ, A. Sistemas de bancos de dados. 3. ed. São Paulo: Makron Books, 1998.

HEUSER, C. A. Projeto de Bancos de Dados. 2. ed. Porto Alegre: Sagra Luzzatto, 1999.

PRESSMAN, R.S. Engenharia de software. São Paulo: Makron Books, 1995.

ROB, P.; CORONEL, C. Sistemas de Banco de Dados: Projeto, Implementação e Administração. 8. Ed. São Paulo: Cengage Learning, 2011.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: Elsevier, 2006.

DATE, C. J.; Introdução a Sistemas de Banco de Dados; 8^a. Ed.; Trad. Daniel Vieira; Rio de Janeiro: Elsevier, 2003.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4 ed. Instituto de Informática da UFRGS, Sagra DC Luzzatto, 1998.

RAMAKRISHNAN, R. GEHRKE, J. Database Management Systems. 2. ed. Boston: McGraw-Hill, 2000.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T.; JAGADISH, H. V.; Database Modeling and Design – Logical Design. 5^a ed., Burlington – USA: Elsevier, 2011.



NO PRÓXIMO CAPÍTULO

No próximo capítulo, estudaremos sobre Projeto de Banco de Dados. Será apresentado os principais detalhes existentes por trás de um projeto de banco de dados bem estruturado, sobretudo, suas principais fases inerentes a modelagem de dados.

2

Projeto de Banco de Dados

2 Projeto de Banco de Dados

No capítulo 1, aprendemos os conceitos de dados e informações, e como são importantes e valiosos para as empresas. Aprendemos que a informação é considerada o principal ativo das empresas, auxiliando na tomada de decisão e, consequentemente, se tornando um fator de competitividade empresarial.

Após abstrairmos a relevância das informações no cotidiano empresarial, estudaremos sobre o Projeto de Banco de Dados, que provê mecanismos eficazes para elaboração e criação de banco de dados organizacionais, permitindo a manipulação correta de dados.

Atualmente, a grande maioria das organizações, independentemente do seu porte (pequeno, médio e grande), impreterivelmente, utilizam SGBDs para gerenciarem suas informações empresariais de maneira segura, íntegra e, ao mesmo tempo, disponibilizá-las para consultas a qualquer momento. Bem, você deve estar se perguntando, afinal, qual é a importância real de um Projeto de Banco de Dados? Quais são suas principais fases? Como projetar um banco de dados conciso e bem estruturado? Não tenha dúvida, que nessa unidade você conhecerá as respostas desses questionamentos. Então, vamos aos estudos!



OBJETIVOS

Este capítulo tem como objetivo expor as principais fases de um projeto de banco de dados. Estudaremos, de maneira objetiva, os modelos externo, conceitual e externo. Conheceremos os principais modelos de dados. É extremamente importante que exploremos profundamente o modelo relacional, apresentando suas principais características e vantagens, se compararmos com os demais modelos de dados presente e disponíveis para utilização.

Prepare-se, pois você terá um trabalho ardo pela frente, então à dica é, dedique-se bastante a este capítulo!



REFLEXÃO

Você se lembra de nossa discussão no capítulo 1 sobre o conceito de dado e informação? Se, eventualmente, alguém perguntar a diferença entre dado e informação, qual será a sua resposta?

2.1 Projeto de banco de dados

Para que alcancemos com êxito os objetivos de um SGBD, torna-se imprescindível obtermos uma estrutura de dados adequada e bem definida, bem como, escolher um modelo de dados para ser implementado pelo banco de dados.

! ATENÇÃO

*Um banco de dados pode ser considerado como uma coleção de dados persistentes, onde esses mesmos dados são manipulados por sua vez por aplicações empresariais (DATE, 2003).

Quando iniciamos a constituição de um projeto de banco de dados, normalmente, torna-se necessário se basear em uma visão abstrata do cenário, o que consideramos também como processo de abstração do mini-mundo, inserindo detalhes à medida que o projeto decorre. Essa maneira de utilizar níveis de abstração de dados promove facilidade para que possamos agrupar as diversas visões de dados existentes dentro das organizações empresariais. Como exemplo, podemos considerar que visão de dados abstraída pela diretoria é plenamente distinta da visão de dados dos funcionários vinculados à produção. Esse detalhe, sem dúvida, deverá ser considerado na fase de projetar o banco de dados de uma organização qualquer, independentemente, da sua área de atuação.

Por isso que, o Comitê de Planejamento e Exigência de Padrões (SPARC) do Instituto Nacional Americano de Padrões (ANSI), em meados de 1970, elaborou uma estrutura cujo objetivo era auxiliar a fase de modelagem de banco de dados baseando-se em diversos níveis de abstração de dados. A arquitetura (ANSI/SPARC) considera apenas três níveis de abstração de dados, o externo, conceitual e interno. Para que você entenda melhor as hierarquias dos modelos de dados, visualize adequadamente a Figura 2.1, ora adaptada, incluindo o modelo físico, o qual lida explicitamente com os detalhes referente ao modelo interno em seu nível físico.

! ATENÇÃO

O American National Standards Institute (ANSI) por meio do *Standards Planning and Requirements Committee* (SPARC) estabeleceu um padrão para o desenvolvimento de tecnologias de banco de dados, definindo uma arquitetura de 3 níveis independentes, a citar: interno, conceitual e externo. (Rob, 2005).

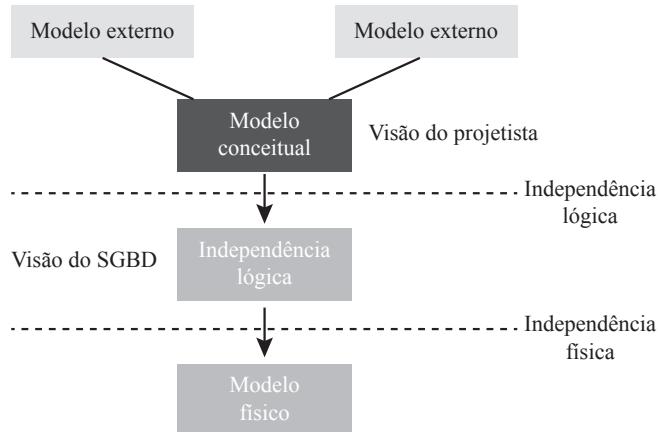


Figura 2.1 – Níveis de abstração de dados

Você pode perceber que a visão do usuário final, isso é, do modelo externo, constitui por sua vez, o modelo conceitual, ou seja, o modelo de dados em que a grande maioria dos projetistas de dados se baseia para a elaboração de qualquer projeto de banco de dados, independentemente de sua complexidade.

É importante mencionar que o nível de abstração (modelo externo/conceitual) é independente de *hardware* e *software* (SGBD). Entretanto, o próximo nível, o modelo lógico, ao contrário do modelo conceitual, depende do software (SGBD), todavia, também não considera o hardware. Para finalizar, nosso último modelo, esse intitulado de modelo interno (modelo físico), depende de maneira exclusiva do *software* (SGBD) e *hardware*.

2.2 Modelo externo

O modelo externo considera o cenário do ambiente de dados de todos os tipos de usuários, porém, em especial, os usuários finais. Recapitulando os conceitos já estudados, consideramos como usuários finais a nomenclatura utilizada aos usuários que utilizam aplicativos computacionais para promover a mani-

pulação dos dados, obtendo como resultado final a produção de informações relevantes para o ambiente empresarial.

Acredito que seja evidente para você que, a maioria das empresas é segmentada em diversos departamentos, como por exemplo: departamento de recursos humanos, financeiro, vendas, etc. Onde cada departamento deverá atender certas restrições e requisitos específicos. Dessa maneira, cada usuário final que desempenha suas atividades nesses diversos departamentos possui uma visão de apenas um subconjunto de dados, específicos do seu departamento, desprezando os detalhes dos outros departamentos.

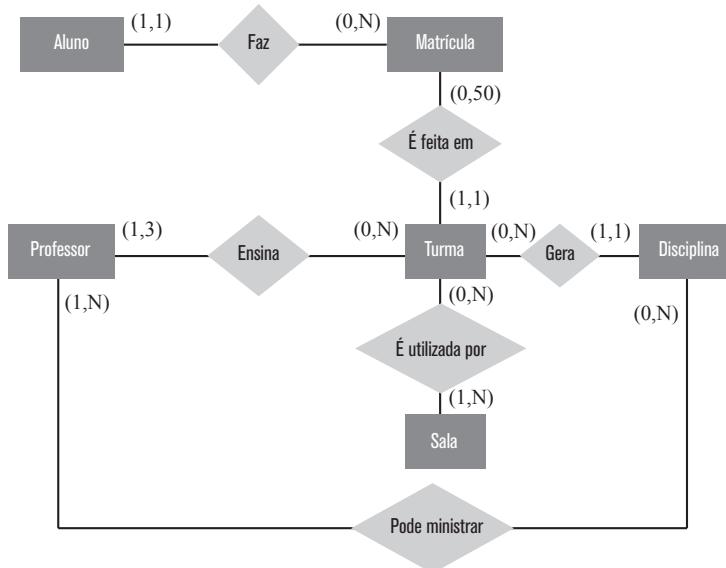


Figura 2.2 – Modelo Externo (controle acadêmico)

Por meio da Figura 2.2 acima, a qual podemos visualizar o modelo externo pertinente ao controle acadêmico de uma universidade fictícia, é factível abstrairmos o modelo externo citado como exemplo, será o responsável em promover o armazenamento dos dados dos alunos, matrículas, professores, disciplinas e turmas. Você pode observar também que as visões das diversas funcionalidades do sistema são, por sua vez, segmentadas entre si, e, cada visão pode também compartilhar um conjunto de dados com a outra (dados dos alunos e cursos sendo compartilhados com a entidade “disciplina”).



Leia um pouco mais sobre a história e evolução dos SGBDs e da SQL:

http://pcleon.if.ufrgs.br/~leon/Livro_3_ed/node116.html

http://algol.dcc.ufla.br/~heitor/Artigos/Artigo_008.html

Você pode ainda observar, por meio da Figura 2.2, que as entidades e seus respectivos relacionamentos nos apresentam alguns detalhes mencionados a seguir:

- Cada disciplina possui, obrigatoriamente, um professor vinculado, e, por sua vez, um professor pode ministrar várias disciplinas, até mesmo, nenhuma;
- Você pode perceber ainda que, apesar dos professores ministrarem várias disciplinas ou nenhuma em uma universidade, dessa forma, o modelo apresenta o uso do que chamamos de cardinalidade mínima entre PROFESSOR e DISCIPLINA como sendo 0 (zero). Sendo assim, admitimos que pode existir disciplinas sem professores vinculados;
- Um aluno pode ou não realizar matrículas, essas vinculadas as turmas, que por sua vez, deverá estar associada a pelo menos 1 disciplina;
- Um professor pode ministrar aula para várias turmas, inclusive nenhuma. De outro lado, uma turma pode possuir no mínimo 1 professor ou no máximo 3 (três);
- Para finalizar, ainda temos a que uma turma utiliza no mínimo 1 sala de aula, que, em contrapartida, pode ser utilizada por várias turmas, inclusive nenhuma.

2.3 Modelo Conceitual

Posteriormente estudaremos os detalhes relacionados ao modelo externo, podemos identificar que utilizamos o modelo conceitual, que por sua vez é representado graficamente pelo diagrama de entidade-relacionamento (DER). Esse DER têm como propósito realizar a integração de todas as visões externas em uma única e simples visão. Dessa forma, o modelo conceitual permite uma abrangente abstração do banco de dados, simplesmente por proporcionar a integração de todas as visões externas (entidade, relacionamentos e, eventuais restrições de domínio e ou referencial) em uma visão única de todos os dados manipulados pela empresa.

O modelo ER (entidade-relacionamento) é considerado um modelo conceitual mais empregado pelos projetistas de bancos de dados. Você não poderá se esquecer de que, o modelo ER é representado graficamente por meio do uso do diagrama de entidade-relacionamento (DER), que por sua vez, representa o esquema de banco de dados.

O modelo conceitual inclui diversas vantagens, entretanto, algumas delas devem ser destacadas:

- Possibilita uma visualização global simples do cenário cujo banco de dados será implementado;
- É um modelo independente da tecnologia do SGBD, ora utilizado para a implementação física;
- Não podemos deixar de destacar que esse modelo também não depende do hardware do servidor de banco de dados, sendo assim, as alterações oriundas do *hardware e software* não afetará o modelo conceitual.



CONEXÃO

Recomendamos a leitura deste artigo, para melhor compreensão do Projeto de Banco de Dados:
www.dpi.ufv.br/~jugurta/papers/tesejug.pdf

2.4 Modelo interno

Quando o projetista de banco de dados chega nessa fase, é imprescindível que o mesmo já tenha escolhido a tecnologia de sistema de gerenciamento de banco de dados (SGBD) que será empregado. O objetivo do modelo interno é realizar o mapeamento do modelo conceitual para um determinado SGBD.

Sendo assim, quando você utiliza um modelo relacional, consequentemente, você também deverá escolher um banco de dados que permita o mapeamento do modelo conceitual para as tabelas (relações) existentes no modelo relacional. Provavelmente, você deve estar interrogando como é formado o modelo interno? Diríamos que essa é uma ótima pergunta! O esquema/modelo interno é formado pela utilização da linguagem SQL (*Structured Query Language*), assim que definimos o SGBD-R (Sistema de Gerenciamento de Banco de Dados Relacional). A fim de elucidar melhor o que efetivamente é realizado nessa fase, a Figura 2.3 a seguir, representa um modelo interno, constituindo as relações (tabelas) aluno, departamento, disciplina e curso.

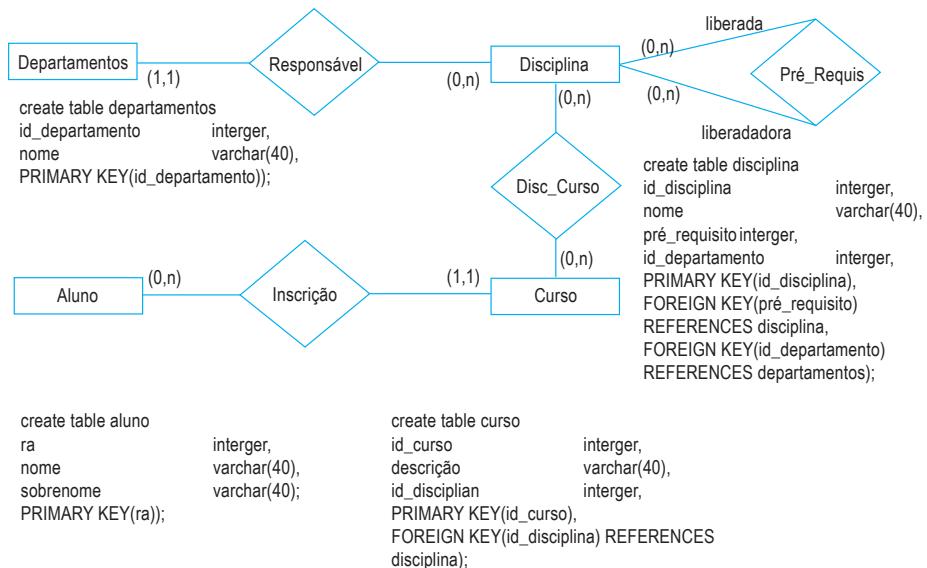


Figura 2.3 – Modelo interno (sistema acadêmico)

Apenas para enfatizar novamente, você já deve ter percebido que o modelo interno está ligado diretamente com a tecnologia do SGBD a ser utilizado, ou seja, existe uma dependência em nível de software, por exemplo, qualquer alteração, por mais sucinta que ela seja, realizada no SGBD demandará que o modelo interno também reflita algum tipo de alteração para que seja possível contemplar certos requisitos/restrições. Eventualmente, chegamos ao ponto de alterarmos o modelo interno sem resvalar no modelo conceitual, caracterizando o que chamamos de independência lógica.

2.5 Modelo físico

Esse modelo é conhecido por trabalhar com o nível mais baixo de abstração, apresentando de maneira detalhada como os dados são efetivamente gravados em um dispositivo de armazenamento qualquer (disco rígido, fitas magnéticas, etc.). O modelo físico depende exclusivamente do SGBD (*software*) e do *hardware*, sobretudo por precisar conhecer previamente os dispositivos físicos que serão utilizados para o armazenamento dos dados, principalmente, os métodos que proverão acesso aos mesmos.



ATENÇÃO

Tuning SQL (otimização de consulta) é um processo de selecionar o plano de avaliação de consulta mais eficiente dentre as muitas estratégias normalmente possíveis para o processamento de determinada consulta (Silberschatz, A. et al, 2006).

O DBA (Administrador de Banco de Dados) e ou projetista de dados não precisam se preocupar com os detalhes em nível do armazenamento físico, entretanto, precisam conhecer minuciosamente como realizar a sintonização (*tuning*) a fim de incrementar a performance, principalmente quando utilizamos o modelo de dados relacional.

Quando for necessário aplicar qualquer tipo de alteração no modelo físico sem precisar refletir a alteração/modificação no modelo interno, obtemos o que caracterizamos de independência física de dados.

2.6 Modelo de dados

O modelo de dados pode ser conceituado como sendo uma representação simples que, normalmente, simboliza graficamente as estruturas dos dados. Podemos dizer que o modelo de dados é uma abstração de um ambiente real, cujo objetivo é subsidiar o compreendimento adequado dos requisitos, sejam esses simples ou complexos, inclusos em cenários empresariais.

O projetista de dados deverá utilizar o bom senso para que seja possível a obtenção de modelos de dados bem estruturados e aceitáveis. O processo relacionado à elaboração de um modelo de dados não é uma tarefa considerada nada fácil, e, certamente uma versão satisfatório do modelo de dados será atingida, posteriormente diversas correções e adaptações.

Para exemplificar, e garantir seu entendimento pleno acerca do conceito do modelo de dados, utilizaremos a turma da disciplina de Modelagem de Dados, a qual deverá ser divida em três grupos distintos, onde cada grupo se responsabilizará em elaborar um modelo de dados para contemplar as necessidades de armazenamento de dados de uma locadora de automóveis (regra de negócio). Você poderá perceber que cada grupo apresentará uma versão de modelo de dados, cujo o propósito é basicamente o mesmo, ou seja, atender os requisitos fundamentais da regra de negócio de uma locadora de automóveis. Agora, qual modelo realmente estará correto? Você pode concluir que não existe uma res-

posta única e exclusiva para esse questionamento. O mais adequado seria responder que: “o melhor modelo de dados é aquele que atende em sua plenitude todos os requisitos da regra de negócio e, certamente, podemos chegar a mais de uma alternativa considerada correta”.

Sumarizando, podemos dizer que os modelos de dados servem como ferramentas de comunicação, fomentando uma abstração global de como os dados serão estruturados e armazenados no novo banco de dados a ser confeccionado.

Analogamente, imagine a abstração de uma planta baixa de uma residência qualquer. Você deve concordar que não conseguirmos morar na planta, ok! Sendo assim, um modelo de dados também pode ser considerado como uma abstração, isto é, não é possível realizar qualquer tipo de manipulação de informação a partir de um modelo de dados. Resumidamente, não podemos levantar uma residência bem estruturada e confiável abdicando-se do uso de uma planta baixa, isso poderá ser aplicado para o banco de dados sem nenhuma restrição, sobretudo por ser imprescindível para a criação de qualquer banco de dados a realização prévia de um modelo de dados.

Os modelos de dados são constituídos pelo emprego de entidades, atributos, relacionamentos e eventuais restrições de domínio ou referencial. Se você estiver confuso, principalmente por ainda não termos elucidado os conceitos acerca de entidade, atributo e relacionamento, não se preocupe, detalharemos na sequência o significado de cada conceito.

Consideraremos como entidade, algo que desejamos armazenar no banco de dados, a citar: um carro, uma pessoa, etc., que por sua vez, representa um determinado tipo de objeto abstraído do mundo real (mini-mundo). Dessa maneira, concluísse que cada ocorrência dessa mesma entidade é considerada única e exclusiva. Como exemplo, podemos tomar como base a criação de uma entidade, ora intitulada de FUNCIONARIO. Certamente teríamos “n” ocorrências de funcionários únicos, como exemplo tomamos os seguintes nomes de funcionários: Geraldo Alckmin, Aécio Neves, Dilma Rousseff, etc. As entidades podem constituir objetos físicos reais (funcionários, clientes, alunos, professores, etc.), como também uma forma de abstração, a citar como exemplo, os horários de uma determinada ponte-aérea Rio de Janeiro-São Paulo ou uma exibição de uma filme específico no cinema.

Por outro lado, um atributo é conceituado como uma característica particular de uma entidade, ou até mesmo de um relacionamento específico. Isso mesmo! Podemos associar um atributo a um relacionamento, não apenas vincula-lo em uma entidade. Para exemplificar o uso de um atributo, suponha que caracteriza-

mos a entidade FUNCIONARIO com os seguintes atributos: número do funcionário, nome, sobrenome, endereço e salário.

Um relacionamento tem como propósito descrever um vínculo (associação) entre uma ou várias entidades. Podemos tomar como exemplo a existência de um relacionamento que por sua vez, associa/vincula as entidades FUNCIONARIO e CLIENTE que pode ser interpretada da seguinte maneira: um funcionário pode atender “n” clientes, todavia, cada cliente pode ser atendido por somente um funcionário. A fim de representar adequadamente esses vínculos, o modelo de dados faz uso de três tipos básicos de relacionamento (o que também chamamos de cardinalidade): um para muitos (1:M ou 1..*), muitos para muitos (M:N ou *..*) e um para um (1:1 ou 1..1).

Ao mencionarmos restrição aplicada ao modelo de dados, podemos entender como sendo uma limitação imposta aos dados. Para aplicar a integridade de dados, torna-se crucial a utilização de restrições. O uso de restrições em um modelo de dados é considerado de suma importância. Para exemplificar o uso de restrições aplicada em um domínio qualquer, considere os detalhes a seguir:

- Um funcionário deverá possuir um salário entre o intervalo de R\$ 780,00 e R\$ 7.500,00;
- Um funcionário deverá possuir como média de venda, um valor superior a 30% de seu salário atual;
- Cada exemplar em uma biblioteca deverá possuir no mínimo 5 livros para realização de empréstimos.

Você já sabe identificar corretamente as entidades, atributos e relacionamentos e ou eventuais restrições aplicada ao domínio? Inicialmente é necessário você absorver corretamente os requisitos da regra de negócio a ser modelada.

Bem, mencionamos diversas vezes o conceito de “regra de negócio”, porém, ainda não contextualizamos. Podemos considerar como uma regra de negócio uma descrição simples, clara e sem imprecisão no que tange as transações ou objetos de uma empresa qualquer, sem considerar o seu porte (pequena, média ou grande). Por meio do uso das regras de negócio podemos identificar as possíveis entidades, atributos, relacionamentos e, eventuais restrições. Não tenha dúvida que, sem perceber, você já se esbarrou em alguma regra de negócio e nem se deu conta. Em nossos exemplos anteriores, quando referimos que “um funcionário pode atender “n” clientes, porém, cada cliente pode ser atendido por um exclusivo funcionário”, essa descrição é considerada como sendo uma regra de negócio.

Agora que já desmitificamos o conceito de “regra de negócio”, podemos apresentar os diversos modelos de dados, ora constituídos para promover o melhor gerenciamento dos dados, que tiveram como objetivo solucionar eventuais falhas provenientes dos antigos sistemas de arquivos. Constituímos uma tabela, essa intitulada de Tabela 2.1, a fim de demonstrar resumidamente os modelos de dados mais habituais, considerando a evolução do tempo:

PERÍODO	MODELO	EXEMPLOS	COMENTÁRIOS
1960 a 1970	Sistema de arquivos	VMS/VSAM	IBM faz uso em seus mainframes Não utiliza relacionamentos
1970	Modelo de dados hierárquico e em rede	IMS, ADABAS, IDS-II	Implementado pelos primeiros bancos de dados da época
1970 até o presente	Modelo de dados relacional	DB2, Oracle, MS SQL Server, MySQL	Utiliza os conceitos básicos (simples) e objetivos.
1980 até o presente	Orientado a objetos Relacional estendido	PostgreSQL, Versant, Caché, FastObjects.Net	Usado na manipulação de dados considerados complexos. Disseminação de banco de dados na web
Do presente ao futuro	XML	Objectivity/DB, DB/2 UDB, dbXML, Tamino, DB2 UDB, Oracle 10g, MS SQL Server, PostgreSQL	Permite a manipulação e gerenciamento de dados semiestruturados. Suporte a documentos no formato XML

Tabela 2.1 – Evolução dos Modelos de Dados (Desde 1960).

2.6.1 modelo hierárquico

O modelo hierárquico foi criado em meados de 1960, com o propósito de gerenciar adequadamente grandes volumes de dados provenientes de projetos complexos. A estrutura lógica do modelo hierárquico é constituída por uma estrutura semelhante a estrutura de uma árvore, essa visualizada de cima para baixo, possibilitando visualizar suas ramificações.

No modelo hierárquico, consideramos um segmento como sendo um registro em um sistema de arquivo. Internamente, o modelo hierárquico possui uma camada superior, essa denominada de raiz (nó pai) do segmento subsequente abaixo desse. A Figura 2.4 exemplifica o uso de um modelo hierárquico, onde é possível identificarmos o segmento “pai” considerado raiz (voo) dos segmentos “filhos” (escala, reserva e bilhete), onde escala e bilhete, por sua vez, são considerados “pais” dos fragmentos cidade e cliente. De maneira intuitiva, os segmentos ora localizados abaixo de outros segmentos são considerados “filhos”. Sendo assim, se observarmos detalhadamente, é factível de identificar que o modelo hierárquico representa exclusivamente o relacionamento um para muitos (1:M) existente entre o segmento pai e seus respectivos filhos, ou seja, cada segmento pai possui diversos segmentos filhos, porém, cada segmento filho, por sua vez, possui apenas vinculado a ele um segmento pai.

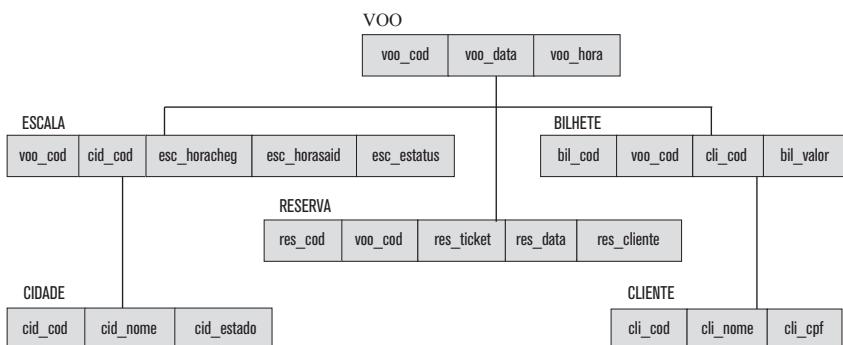


Figura 2.4 – Exemplo fictício de um Modelo Hierárquico

Já na década de 70, o banco de dados hierárquico era referência, sobretudo por possuir grandes vantagens sobre os sistemas de arquivos, constituindo assim, pilares fundamentais para subsidiar o desenvolvimento de aplicativos comerciais. Mesmo o modelo hierárquico possuindo diversas vantagens em com-

paração ao armazenamento de dados por meio do uso de sistema de arquivo, esse modelo apresentava limitações relevantes, as quais, podemos mencionar como principais a ausência de independência estrutural, dificuldade em gerenciar e manipular os registros, e, a maioria dos relacionamentos de dados não conseguiam se adaptar à cardinalidade (multiplicidade) 1:M.

2.6.2 modelo em rede

O modelo de rede foi constituído com objetivo de representar os relacionamentos de dados considerados mais elaborados, adotando uma representação simples e eficaz em comparação ao modelo hierárquico.

Ao contrário do modelo hierárquico, o modelo em rede possibilita que um registro possua mais de um segmento pai, conforme podemos visualizar na Figura 2.5 abaixo apresentada:

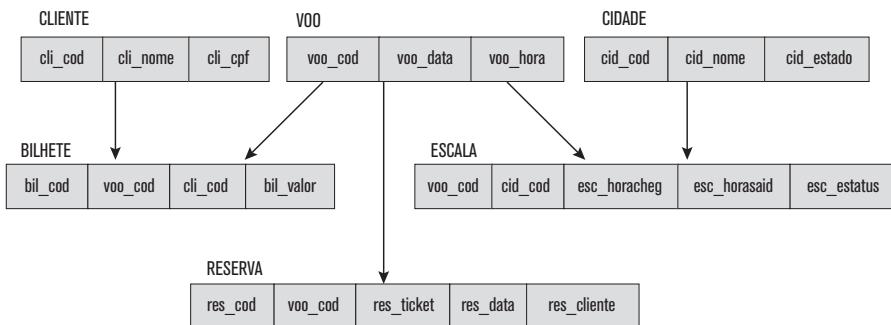


Figura 2.5 – Modelo em Rede (exemplo fictício).

Nesse exemplo, simplesmente convertemos o modelo hierárquico para o modelo em rede, utilizando as mesmas regras de negócio. Você já deve ter identificado os tipos de registros, a citar: CLIENTE, VOO, CIDADE, BILHETE, ESCALA e RESERVA. Também é notório que ESCALA é considerado filho de ambos, isso é, de voo e cidade, e que, esses, por sua vez são denominados pais. Análogo a descrição anterior, podemos dizer que bilhete também possui dois pais (cliente e voo).

Entretanto, o modelo em rede não conseguiu solucionar todos os problemas pertinentes à modelagem de dados. A ausência de estrutura correta para permitir a realização de consultas e a necessidade de implementação

maciça de linhas de código-fonte para fornecer um simples relatório era considerada uma das grandes desvantagens desse modelo de dados. A falta de independência de dados também era considerada um de seus pontos negativos, pois, caso fosse necessário realizar qualquer tipo de modificação que atingisse a estrutura dos dados, por mais sucinta que fosse correríamos o risco de destruir os aplicativos que utilizassem o banco de dados para a manipulação dos dados.

Devido essas inúmeras desvantagens correlacionadas anteriormente, ora apresentadas pelo modelo hierárquico e em rede, por volta da década de 80, surgiu o modelo de dados, esse intitulado de modelo relacional, que acabou por substitui-los.

2.6.3 modelo relacional

O modelo relacional foi criado proveniente do conceito matemático (relação). Esse conceito nos permite abstrair que uma relação nada mais é do que uma matriz bidimensional, ora constituída por linhas e colunas.

Esse modelo, frequentemente, é constituído por meio do uso de um sistema gerenciador de banco de dados relacional, ora referenciado pela sigla SGBD-R. O SGBD-R dispõe das mesmas funcionalidades apresentadas pelos sistemas gerenciador de banco de dados que atende ao modelo hierárquico e em rede, excluindo a inserção dos demais recursos os quais permitem que o modelo relacional acrescente maior complexidade em sua abstração e implementação.

Ao mencionarmos as vantagens de um SGBD-R, é importante correlatar a maneira em que o mesmo oculta a complexidade do modelo relacional, administrando os detalhes físicos e expondo o banco de dados relacional aos usuários finais como sendo um conjunto de relações (tabelas) onde os dados são armazenados, essas se relacionando entre si.

No modelo relacional, quando compartilhamos um determinado atributo de uma tabela específica, torna-se possível promover o relacionamento entre si.

Para exemplificar, apresentamos a Figura 2.6, ora constituída pelas tabelas nomeada de CLIENTE, essa por sua vez, possui o número do funcionário (ID_FUNC) que normalmente lhe atende, vinculado pela tabela FUNCIONARIO.

Tabela: Funcionário

ID_FUNC	NOME	SOBRE-NOME	SEXO	CPF	RG
987	José	Abrão	M	111222333444	1234567890
321	Márcia	Marina	F	555333123098	2347698243
112	Wagner	Moura	M	765234509876	123543-MG

Relacionamento por meio do atributo ID_FUNC

Tabela: Cliente

ID_CLI	NOME	SOBRE-NOME	SEXO	CPF	RG	ID_FUNC
89710	Geraldo	Alckmin	M	7687171398	1233456	112
89711	Aécio	Neves	M	6534982615	2455455	987
65412	João	Gilberto	M	1235566615	2342353	987
23113	Marina	Silva	F	6509090456	4563434	112
65514	Débora	Santos	F	1236451324	3425633	112

Figura 2.6 – Exemplo de Modelo Relacional

É factível de identificarmos o estabelecimento do relacionamento existente entre as relações FUNCIONARIO e CLIENTE que permite vincular um cliente a um determinado funcionário, esse responsável pelo atendimento. Repare que,

mesmo que os dados dos clientes estejam armazenados em uma relação e, os dados dos funcionários, por sua vez, armazenados em outra, é possível trabalharmos com essa integridade referencial.

O exemplo exposto acima é suficiente para o seu aprendizado no que se refere ao modelo relacional? Ainda lhe resta dúvida? Bem, vamos estudar um pouco mais! Podemos vincular os clientes Geraldo Alckmin, Marina Silva e Débora Santos com seu respectivo vendedor, o funcionário chamado Wagner Moura, esse identificado exclusivamente pelo número 112. Você pode identificar ainda que, na tabela CLIENTE, o atributo rotulado de ID_FUNC é por sua vez uma chave estrangeira, a qual associa os clientes com seus respectivos vendedores.

Dessa forma, concluísse que o modelo relacional, na trajetória da evolução dos modelos de dados, foi considerado imprescindível, principalmente por incorporar a linguagem SQL (*Structured Query Language*), que por si só nos proporciona total transparência na manipulação de dados e confecção de relatórios gerenciais.



ATIVIDADE

4. Descreva detalhadamente o conceito de Entidade e Relacionamento. Cite pelo menos três exemplos onde podemos utilizar ambos.
5. Analise o cenário do ambiente acadêmico, mais especificamente, de uma sala de aula. A partir dessa análise, represente por meio de um DER, o conjunto de carteiras e o conjunto dos tipos de móveis.
6. Discorra sobre os detalhes pertinentes ao Modelo Hierárquico, apontando suas desvantagens comparando com o Modelo em Rede.
7. Realize uma pesquisa na Internet e descreva três características de um Banco de Dados XML. Cite pelo menos três nomes de banco de dados que manipulam arquivos XML.
8. Dê pelo menos três exemplos de restrição aplicada a um modelo de dados. Na sequência, descreva os três tipos de relacionamentos que podem ser utilizado para associar entidades.



REFLEXÃO

Finalizamos mais um capítulo!

Neste capítulo aprendemos sobre as fases que constituem o projeto de banco de dados. Aprendemos que o modelo externo considera o cenário de dados utilizado pelos usuários finais. Estudamos também sobre o modelo conceitual, um dos modelos de dados mais utilizado pelos projetistas de banco de dados (ou DBAs), na elaboração de esquema de banco de dados, o qual é representado graficamente por meio do uso do diagrama de entidade-relacionamento (DER).

Verificamos também que o modelo relacional baseasse no conceito matemático “relação” considerando uma tabela (relação) como sendo uma matriz bidimensional constituída por linhas e colunas.



LEITURA

Artigos on-line: para você incrementar mais o seu nível de aprendizagem relativo ao Projeto de Banco de Dados:

http://juliobattisti.com.br/artigos/office/modelorelacional_p5.asp

<http://www.dicasdeprogramacao.com.br/como-criar-um-projeto-de-banco-de-dados/>

<http://www.devmedia.com.br/dbdesigner-modelagem-e-implementacao-de-banco-de-dados/30897>

Livro: Introdução a Sistemas de Banco de Dados; 8^a. Ed., DATE, C. J, você encontrará mais conceitos acerca de SGBDs, sua utilização e vantagens. Estude mais sobre a arquitetura dos SGBDs, este livro faz uma abordagem bastante densa e completa do assunto. Aprofunde seus conhecimentos!



REFERÊNCIAS BIBLIOGRÁFICAS

ELMASRI, R.; NAVATHE, S.B. Sistemas de bancos de dados. São Paulo: Pearson (Addison Wesley), 2005.

KORTH, H.; SILBERCHATZ, A. Sistemas de bancos de dados. 3. ed. São Paulo: Makron Books, 1998.

HEUSER, C. A. Projeto de Bancos de Dados. 2. ed. Porto Alegre: Sagra Luzzatto, 1999.

PRESSMAN, R.S. Engenharia de software. São Paulo: Makron Books, 1995.

ROB, P.; CORONEL, C. Sistemas de Banco de Dados: Projeto, Implementação e Administração. 8. Ed. São Paulo: Cengage Learning, 2011.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: Elsevier, 2006.

DATE, C. J.; Introdução a Sistemas de Banco de Dados; 8^a. Ed.; Trad. Daniel Vieira; Rio de Janeiro: Elsevier, 2003.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4 ed. Instituto de Informática da UFRGS, Sagra DC Luzzatto, 1998.

RAMAKRISHNAN, R. GEHRKE, J. Database Management Systems. 2. ed. Boston: McGraw-Hill, 2000.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T.; JAGADISH, H. V.; Database Modeling and Design – Logical Design. 5^a ed., Burlington – USA: Elsevier, 2011.



NO PRÓXIMO CAPÍTULO

No próximo capítulo, estudaremos com detalhes o modelo entidade-relacionamento. Estudaremos a importância de constituirmos um modelo conceitual bem estruturado. Para elaborar um bom projeto de banco de dados, devemos aprender as características relevantes do Modelo Entidade-Relacionamento.

3

Modelo Entidade -Relacionamento

3 Modelo Entidade-Relacionamento

No capítulo anterior, estudamos acerca dos conceitos pertinentes ao projeto de banco de dados, discorrendo sobre seus principais níveis de abstração (modelo externo, conceitual e interno). Aprendemos o que efetivamente é realizado no modelo físico e quais são os modelos de dados mais habituais (modelo hierárquico, em rede e relacional).

Espero que você esteja alinhado com todos esses conceitos apresentados até o presente momento, caso esteja com alguma dúvida, sugerimos que você realize uma revisão das unidades anteriores!

Nesse capítulo, estudaremos conceitos triviais para que possamos elaborar um banco de dados utilizando-se o modelo entidade-relacionamento.

Antes de introduzirmos os novos conceitos referentes ao modelo entidade-relacionamento torna-se importante e de grande relevância aprendermos adequadamente sobre as fases constituintes de um projeto de banco de dados, sobretudo, o que é e, para que utilizamos um modelo de banco de dados.

Você está pronto?! Vamos dar início aos nossos objetos de aprendizagem sugeridos neste capítulo 3?



OBJETIVOS

Este capítulo tem como objetivo aprofundar os conceitos pertinentes as principais características do Modelo Conceitual; apresentar o Modelo Entidade-Relacionamento, evidenciando as entidades, relacionamentos, cardinalidade, atributos, dentre outros conceitos; entender como o Modelo de Entidade-Relacionamento é representado graficamente pelo emprego de um Diagrama Entidade-Relacionamento. Compreender como identificar corretamente o grau de um determinado relacionamento seja esse relacionamento unário, binário, ternário e ou “n”ário.



REFLEXÃO

Você se lembra dos conceitos estudados na Unidade 2, onde discutimos sobre o Projeto de Banco de Dados e os principais modelos de dados, e os principais tipos de relacionamentos? Você seria capaz de definir as principais características de um modelo interno de dados? Você se encontra confortável em distinguir adequadamente os conceitos acerca dos Modelos de Dados: Hierárquico, em Rede e Relacional?

3.1 Modelo Entidade-Relacionamento

No capítulo anterior, aprendemos sobre os conceitos básicos de um projeto de banco de dados, seus principais níveis de abstração (modelo externo, conceitual e interno). Aprendemos também como elaborar um modelo físico de banco de dados e discorremos superficialmente sobre os modelos de dados mais comuns (modelo hierárquico, em rede e relacional).

Nesse capítulo, estudaremos os conceitos básicos e fundamentais para a criação de um banco de dados, independentemente da regra de negócio, por meio do uso do modelo entidade-relacionamento.



ATENÇÃO

O processo de modelagem de dados pode ser considerado como sendo um processo que emprega diversas etapas, normalmente iterativo e progressivo. Dá início através de uma abstração simples de um determinado problema, o qual desejamos solucionar, e conforme o nível de abstração do problema é incrementado, o nível de detalhes que a modelagem comprehende também aumenta (Heuser, 2004).

Bem, antes de iniciarmos a discussão sobre os conceitos novos, referente ao modelo entidade-relacionamento, torna-se trivial e ao mesmo tempo, apropriado, obtermos o conhecimento das fases que formam um projeto de banco de dados considerado bem estruturado, e ainda, por que utilizamos o modelo de banco de dados.

Precisamos deixar claro que a fase que constitui a modelagem de dados é a primeira envolvida na criação de um projeto de banco de dados. Essa etapa é simplesmente um processo de criação de um modelo de dados específico que tem como propósito a resolução de um determinado problema, esse presente em nossas atividades diárias.

Reflita: Você faz ideia do tipo de problema que a modelagem de dados se propõe a solucionar?

Não tenha dúvida que essa resposta é bem simples! Certamente todos nós nos deparamos com eventuais problemas em nosso cotidiano, isso se replica, as organizações empresarias. Normalmente, tais problemas são claramente definidos no ambiente empresarial real, com escopo e limites bem delimitados, que por sua vez, devem ser tratadas como uma visão sistêmica.

Você imagina qual seria o tipo de problema que a modelagem de dados se propõe a solucionar?

Para exemplificar, imagine a necessidade de realizar a categorização de um conjunto de produtos. Dizemos que processo de categorizar esse conjunto de produtos seja o nosso problema! Para que posamos promover uma modelagem de dados coerente e concisa, devemos realizar criteriosamente o levantamento de todos os detalhes e, eventuais relações que isto implicaria dentro de uma regra de negócio específica.



ATENÇÃO

Um esquema de banco de dados nada mais é do que uma representação de um modelo de dados por meio do uso de um conceito de modelagem de dados (DATE, 2003).

3.1.1 Modelo de Banco de Dados

Segundo Peter Chen, um **Modelo Entidade-Relacionamento** é constituído por uma notação, ora formada por entidades (graficamente representado por retângulos), relacionamentos (graficamente representado por losango), atributos, que compõe as características das entidades, e ou, relacionamentos (esse, graficamente representado pelo emprego de círculos) e, por fim, linhas que realizam a conexão indicando por sua vez a cardinalidade (multiplicidade) de uma entidade ou mais, aplicado a um relacionamento qualquer. Essas conexões são graficamente representadas pelo uso de linhas.

Até o momento, aprendemos que um modelo de banco de dados nada mais é do que uma descrição repleta de detalhes acerca das principais informações que almejamos armazenar em um banco de dados qualquer. Dessa maneira, é crucial adotarmos uma linguagem para realizar essa modelagem de dados, conduzindo-nos a confeccionar modelos de dados confiáveis e estruturado.

Classificamos essas linguagens de acordo com a representação dos modelos de dados. Como exemplo, podemos correlacionar, linguagens que utilizam textos ou gráficos para realizar a representação dos modelos de dados em distintos níveis de abstração.

Dessa forma, dizemos que um esquema de banco de dados nada mais é do que uma representação de um modelo de dados, que utilizou uma linguagem específica de modelagem de dados.

Um modelo de dados tem como principal característica ser simples no que se refere ao entendimento de como os dados serão organizados e manipulados em um banco de dados, mesmo para aqueles usuários desprovidos de conhecimentos computacionais específicos.

Ao iniciarmos a elaboração de um projeto de banco de dados, é aconselhável que tenhamos pelo menos dois níveis de abstração, esses, respectivamente, nomeados de modelo conceitual (projeto conceitual) e modelo lógico (projeto lógico).

3.1.2 Modelo Conceitual

O modelo conceitual tem como propósito descrever um modelo de dados de forma abstrata e independente da tecnologia do SGBD empregado. Esse modelo também descreve genericamente, quais dados deverão ser armazenados no banco de dados, entretanto, não menciona como estes mesmos dados serão armazenados em nível de software (SGBD).

Essa abordagem é amplamente conhecida como Entidade-Relacionamento (ER), que por sua vez, é considerada uma das principais técnicas utilizadas na modelagem conceitual. Tal técnica nos permite a representação gráfica do modelo conceitual através de um diagrama, esse, intitulado de diagrama entidade-relacionamento (DER). A Figura 3.1 nos apresenta um exemplo do diagrama entidade-relacionamento:

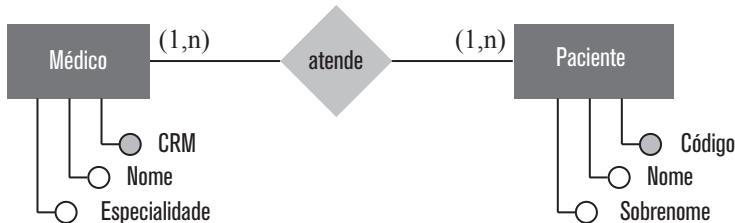


Figura 3.1 – Representação do modelo conceitual por meio de um DER

O modelo conceitual, ora representado graficamente pelo emprego do diagrama entidade-relacionamento, permite abstrair que o banco de dados conterá dados referentes aos médicos e pacientes. Dessa maneira, para cada médico, o banco de dados armazenará suas principais características, a citar, CRM (número do Conselho Regional de Medicina), nome e sobrenome, e, por sua

vez, para cada paciente, serão armazenados seu código identificador, nome e sobrenome. Ainda podemos identificar a existência de uma associação (relacionamento) ora aplicada para descrever os atendimentos médicos.

3.1.3 modelo lógico

O modelo lógico pode ser considerado o oposto do modelo conceitual, visto que o mesmo depende exclusivamente do SGBD que será utilizado na criação do banco de dados. Esse modelo tem como objetivo descrever o banco de dados em um nível de abstração visto pelo usuário do SGBD.

Na fase constituinte do modelo lógico, o projetista de dados decidirá as tabelas (relações) que formarão o banco de dados e, ainda, definir para cada tabela, os respectivos nomes de suas colunas, por exemplo: tabela intitulada de “médico” será constituída pelas colunas (atributos) CRM, nome e sobrenome. Já por sua vez, a tabela nomeada de “paciente” será formada pelas colunas COD_PACIENTE, nome, sobrenome e CRM.

Esses detalhes, ora visualizados pelo usuário do SGBD correspondem ao armazenamento interno dos dados, que normalmente, interferem no desempenho da aplicação computacional.

Na primeira fase do projeto de banco de dados, uma das principais atividades é o que denominados de levantamento de requisitos, atendendo as necessidades organizacionais de um ambiente empresarial qualquer, através da análise de requisitos, criando assim, subsídios para a elaboração do projeto conceitual, ora representado pelo modelo entidade-relacionamento (MER). O MER trata-se de um modelo de dados considerado de alto nível de abstração, que independe da tecnologia do SGBD adotado para a realização do armazenamento dos dados.



CONEXÃO

Leia mais sobre modelo de dados:

<http://www.linhadecodigo.com.br/artigo/332/planeje-o-seu-modelo-de-dados.aspx>

A fase subsequente é destinada a criação do projeto lógico de dados, que possui como propósito realizar o mapeamento do modelo entidade-relacionamento (MER) para um modelo de dados qualquer, a citar, o modelo relacional, ora, obrigatoriamente deverá ser suportado pelo SGBD a ser empregado.

Finalizando o projeto de banco de dados, a terceira fase trata do projeto físico. Esse tem como objetivo definir corretamente as estruturas de armazenamento interno, como podemos mencionar, as tabelas, os índices, dentre outros objetos, ainda sim, definir outras atividades associadas paralelamente, ao desenvolvimento de aplicativos computacionais.

3.2 As Principais Características do MER

Bem, agora chegou o momento de esmiuçarmos as principais características do modelo entidade-relacionamento (MER). Peter Chen, na década de 70, constituiu o modelo entidade-relacionamento, o qual, atualmente é considerado clássico (padrão) para a modelagem conceitual de banco de dados. O objetivo principal do MER é criar adequadamente as entidades e seus respectivos relacionamentos, ora abstraídos de um ambiente empresarial real qualquer, o qual desejamos modelar.

O Modelo Entidade-Relacionamento (MER) descreve conceitualmente como os dados serão manipulados por meio de um sistema computacional.

Uma vasta gama de conceitos é aplicada ao MER, porém, esses conceitos são considerados simples de entender, facilitando consideravelmente as tarefas dos projetistas de dados no que se refere ao entendimento adequado dos conceitos referente aos dados utilizados nos aplicativos computacionais, independentemente da tecnologia do SGBD que será utilizada na implementação do banco de dados. O Diagrama Entidade-Relacionamento (DER), por sua vez, é considerado como sendo um esquema conceitual, ora elaborado a partir dos conceitos do modelo entidade-relacionamento.

Devemos também deixar evidente de que, mesmo que o diagrama entidade-relacionamento possua um elemento chamado de entidade, ora representado graficamente pelo uso de um retângulo, tal elemento não possui nenhuma relação com a entidade externa, essa representado por sua vez, por um diagrama de fluxo de dados.

3.2.1 Entidade

Uma entidade tem como propósito representar um conjunto de objetos de um ambiente organizacional real qualquer, ora a ser modelado. Como exemplo, podemos tomar um funcionário chamado de Willian Bonner, inscrito no RG sob o número 12.345.678-9. O número do RG de Willian Bonner é considerado

como sendo uma entidade, pelo simples fato de permitir identificar exclusivamente um determinado funcionário em comparação com os demais. Destaca-se ainda que uma entidade possa assumir duas características, isso é ser concreta (uma pessoa “funcionário”) e ou abstrata (uma instituição acadêmica).

Dessa maneira, você pode perceber que um conjunto de entidades por sua vez, forma um agrupamento de entidade de um mesmo tipo. Esses conceitos lhe deixaram confuso? Não se preocupe, tentaremos explicá-los para a sua melhor aprendizagem. Suponha um conjunto de funcionários de uma empresa qualquer, por exemplo, a *Oracle*, sendo assim, é permitido definirmos esse conjunto de funcionários como um conjunto de entidades, essas intituladas de FUNCIONARIO.

A Figura 3.2 a seguir, representamos graficamente suas entidades por meio do uso de um diagrama entidade-relacionamento (DER):



Figura 3.2 – Exemplo de entidades representada graficamente pelo DER

Para uma entidade, é sempre aconselhável a existência de um atributo identificador, esse por sua vez, pode ser simples ou composto. O atributo identificador, como o próprio nome diz, permite a identificação única e exclusiva de uma ocorrência de entidade. Todavia, em casos especiais, isso é, em tipos específicos de entidade, não encontraremos atributos identificadores. Esse tipo de entidade, a qual não carece do uso de um atributo identificador é chamado e entidade fraca.

CONEXÃO

Leia mais sobre Diagrama Entidade-Relacionamento (DER):

<http://imasters.com.br/artigo/8568/banco-de-dados/documentacao-de-projetos-web-der/>

Uma entidade fraca depende impreterivelmente da existência de outra entidade, dessa forma, sua existência está vinculada a existência de outra entidade. A identificação exclusiva de uma entidade fraca é imposta pela associação (mesclagem) do atributo identificador da entidade considerada proprietária, e de sua chave parcial.

Eventualmente, em casos pontuais, uma entidade fraca pode vir a ser substituída pelo uso de atributos multivalorados.

Na Figura 3.3, você poderá visualizar um exemplo de entidade caracterizada como fraca. A entidade identificada como DEPENDENTE é a nossa entidade fraca, sobretudo por depender da existência de um FUNCIONARIO. Repare que até a representação gráfica torna-se distinta (retângulo com linhas duplas e ou linha responsável pela conexão com o relacionamento “possui” é mais espessa), a fim de destacarmos a entidade fraca em um DER:

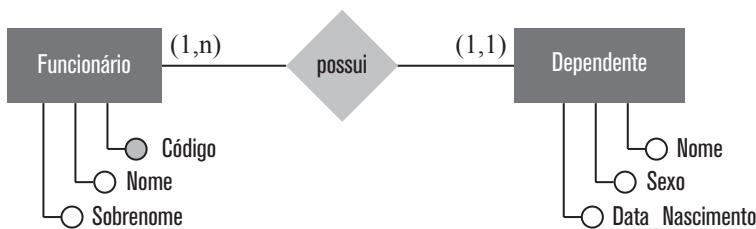


Figura 3.3 – Entidade Fraca (Dependente)

3.2.2 Relacionamento

O uso do relacionamento nos permite realizar associações entre as entidades. Por exemplo, não basta simplesmente conhecermos os funcionários de uma determinada empresa, o projetista de dados deverá associar um funcionário a uma empresa, permitindo que seja possível alcançar algum tipo de informação mais elaborada.



ATENÇÃO

Uma entidade pode ser considerada como sendo um objeto do mundo real ora podendo ser identificado de maneira única. Esse objeto pode ser um elemento concreto, um evento, um ser, uma especialização, uma funcionalidade ou qualquer outro elemento, tangível ou não, do ambiente a ser analisado (CASTRO, E. B. 2012).

No diagrama entidade-relacionamento, um relacionamento é representado graficamente através do uso de um losango, ora conectado por meio de “linhas” as entidades (retângulos). A Figura 3.4 a seguir nos permite visualizar um exemplo de relacionamento, ora identificado de “trabalha”, representando o vínculo existente entre as entidades FUNCIONARIO e EMPRESA.



Figura 3.4 – Exemplo de uso de relacionamento “trabalha”

Vamos agora analisar esse relacionamento, visualizado pela Figura 3.4, que estabelece uma associação entre as entidades FUNCIONARIO e EMPRESA. Ele nos permite referenciar as associações entre o conjunto de entidades. Isso é, para o relacionamento “trabalha”, uma ocorrência pode ser caracterizada como um par de ocorrências ora formada pelas ocorrências das entidades FUNCIONARIO e EMPRESA.

É importante destacar que uma ocorrência de relacionamento não ocorre apenas entre entidades diferentes. O conceito de relacionamento nos permite associarmos ocorrências entre uma mesma entidade, formando o que chamamos de auto-relacionamento (recursivo), como podemos observar na Figura 3.5 a seguir:

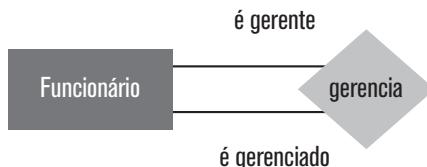


Figura 3.5 – Exemplo de um auto-relacionamento (recursivo)

Perfeito! A Figura 3.5 nos apresentou um exemplo do uso de um auto-relacionamento, existente entre a entidade FUNCIONARIO. Entretanto, antes de continuarmos, é importante discorrermos sobre o conceito de uso de papéis em uma auto-relacionamento.

O uso de “papel” em uma entidade vinculada em um auto-relacionamento tem como objetivo promover a identificação correta de uma instância da entidade dentro de uma instância do relacionamento, ou seja, uma ocorrência de funcionário poderá desempenhar o papel de “é gerente” e a outra ocorrência de funcionário, por sua vez, poderá assumir o papel de “é gerenciado”.

3.2.3 cardinalidade

Em algumas literaturas, podemos encontrar o termo cardinalidade sendo referenciado como multiplicidade. Uma cardinalidade pode ser vista como sendo um exemplo de restrição existente em um diagrama entidade-relacionamento (DER) a fim de atender adequadamente as eventuais exigências do banco de dados.

Para ilustrar um exemplo aplicado a Figura 3.6, observe as cardinalidades máximas no DER abaixo:



Figura 3.6 – Exemplo do uso da cardinalidade máxima

Na sequência, torna-se possível realizarmos a interpretação da cardinalidade máxima imposta sobre as entidades nomeadas de FUNCIONARIO e EMPRESA.

A entidade FUNCIONARIO apresenta uma cardinalidade máxima ilimitada (muitos), essa podendo ser representada pelas letras “N” ou “M”. Dessa maneira, uma empresa pode ter até “N” funcionários trabalhando nela. Por outro lado, a entidade EMPRESA apresenta uma cardinalidade máxima “2”, por sua vez, nos habilita interpretar que um FUNCIONARIO pode trabalhar em no máximo “2” empresas.

Considere o próximo exemplo, esse representado pela Figura 3.7. Repare que utilizamos simultaneamente a representada das cardinalidades mínima e máxima. Observe também que um FUNCIONARIO pode coordenar no máximo 2 (dois) projetos, e por sua vez, um PROJETO pode ser coordenado por no máximo um funcionário.



Figura 3.7 – Utilização das cardinalidades máxima/mínima

Agora que realizamos uma breve apresentação do uso de cardinalidades máxima e mínima, é possível descrever os principais tipos de cardinalidades aplicados em relacionamentos binários. Um detalhe, não se esqueça de que, é possível aplicar esses mesmos tipos de cardinalidades em um auto-relacionamento, isso é, em um relacionamento formado por uma única entidade (relacionamento unário).

O primeiro tipo de cardinalidade é um-para-um, por exemplo, um empregado gerencia apenas um departamento, que por sua vez, é gerenciado por um único empregado, conforme visualizado na Figura 3.8 abaixo:



Figura 3.8 – Cardinalidade 1:1

O segundo tipo de cardinalidade é um-para-muitos, para exemplificar, considere que um empregado gerencia apenas um departamento; todavia, um departamento pode ser gerenciado por muitos empregados, conforme apresentado na Figura 3.9 a seguir:



Figura 3.9 – Cardinalidade 1:N

Já o terceiro tipo de cardinalidade a ser estudada, refere-se a muitos-para-muitos, como exemplo, considere que um empregado gerencia muitos departamentos, e que, por sua vez, um departamento pode ser gerenciado por muitos empregados. Esse exemplo pode ser visualizado por meio da Figura 3.10:



Figura 3.10 – Cardinalidade N:N

Você já deve ter percebido que até o presente momento, no que tange a exemplos de cardinalidade, utilizamos apenas relacionamentos binários, isso é, relacionamentos com apenas duas entidades envolvidas. Não existe nenhum problema quanto a isso, sobretudo por podermos utilizar o mesmo conceito de cardinalidade (multiplicidade) em outros graus de relacionamentos, a citar como exemplo, o relacionamento ternário, ora representado pela Figura 3.11 abaixo:

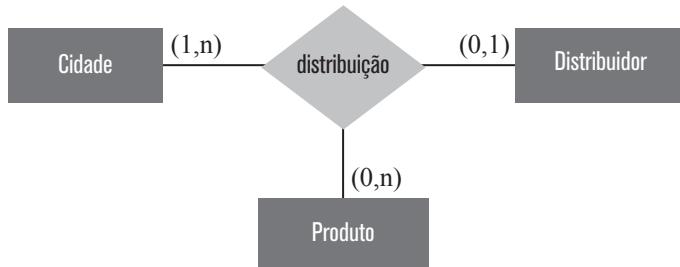


Figura 3.11 – Cardinalidade em relacionamento ternário

Mediante esse exemplo acima apresentado, vamos realizar a devida interpretação para elucidar eventuais dúvidas pertinentes ao conceito de cardinalidade. Cada ocorrência do relacionamento intitulado de “distribuição” vincula três ocorrências de entidade, ou seja, um produto poderá ser distribuído, em uma determinada cidade, onde é realizada a distribuição, por um único e exclusivo distribuidor.

Bem, para clarear ainda mais esse conceito, quando nos deparamos com relacionamentos de duas (binário), três (ternário), ou “n” (“n”ário) entidades, a cardinalidade a ser aplicada trabalha de maneira análoga a cardinalidade aplicada em relacionamentos binários, isso é, torna-se necessário trabalharmos com pares de entidades.

No exemplo da Figura 3.11, dizemos que as ocorrências provenientes das entidades CIDADE e PRODUTO estão associadas a, no máximo uma ocorrência de DISTRIBUIDOR.

3.2.4 Atributo

No modelo entidade-relacionamento MER, é possível realizar a especificação de propriedades relacionadas às entidades. Essas propriedades são nomeadas de atributo. Um atributo promove mecanismos para que seja possível associar informações a ocorrências de entidades e/ou relacionamentos. Resumidamente, um atributo tem como propósito vincular um determinado dado a cada ocorrência de uma entidade específica ou, eventualmente, até mesmo a um relacionamento.

Após apresentarmos o conceito de um atributo, podemos estender nosso aprendizado sobre os vários tipos de atributos existentes, como também apresentar suas devidas utilização. Os detalhes são apresentados logo a seguir:

- **Atributo Simples:** tipo de atributo que armazena um dado atômico, ou seja, um dado considerado indivisível;
- **Atributo Composto:** considerado como sendo um tipo especial de atributo, pelo simples fato de permitir que sejam vinculados a ele diversos dados segmentados de forma isolada por meio de outros atributos. Similar ao atributo simples, um atributo composto também carece de que o dado seja atômico. Como exemplo, podemos mencionar o atributo endereço, pois, o mesmo possui vários dados, como tipo de logradouro, o próprio logradouro, número, complemento, bairro, cidade, CEP e estado;
- **Atributo Multivalorado:** esse atributo permite armazenar vários dados para uma única entidade. Um exemplo clássico para o uso desse tipo de atributo é o número de telefone de uma determinada pessoa. Atualmente, uma pessoa pode ter dois ou mais números de telefones, a citar, telefone fixo e telefone celular de mais de uma operadora;

- Atributo Derivado: o atributo derivado armazena um dado proveniente de um processamento específico. Por exemplo, o atributo idade de uma pessoa qualquer pode ser obtido a partir do processamento da data de nascimento e data atual do aplicativo computacional;
- Atributo Identificador: tipo de atributo imprescindível em uma entidade. Esse atributo permite que identifiquemos exclusivamente uma ocorrência de entidade. O atributo identificador aplicado a uma entidade qualquer pode ser um atributo simples ou composto. Para exemplificar, suponha a existência de uma entidade chamada de Pessoa. Você consegue imaginar os possíveis atributos identificadores que poderíamos utilizar para essa entidade? Como possíveis respostas, teríamos o número do CPF, RG ou um número identificador produzido automaticamente pelo banco de dados.

3.3 Modelo Entidade-Relacionamento Estendido

No modelo entidade-relacionamento estendido apreenderemos como identificar eventuais cenários os quais o uso apenas dos conceitos de entidade e relacionamento visto até o momento não é suficiente para a realização da modelagem de dados. Nesse tópico aprenderemos como utilizarmos agregação, generalização e especialização de entidades como também elaborar um banco de dados completo com o uso do modelo entidade-relacionamento estendido.

3.3.1 Entidade Especializada

É importante mencionar que o modelo entidade-relacionamento estendido contempla todos os conceitos de modelagem apresentados no modelo entidade-relacionamento, incluindo novos conceitos sobre subclasse e superclasse como ainda, os conceitos pertinentes a especialização e generalização.

O nosso primeiro conceito referente ao modelo entidade-relacionamento estendido discorre sobre a subclasse, que por sua vez, refere-se a um determinado tipo de entidade ora utilizada para contemplar uma entidade específica e ou ainda, uma coleção de entidades que eventualmente podemos encontrar em um esquema de banco de dados.

Com o objetivo de exemplificar esse conceito de entidade especializada, considere a entidade FUNCIONARIO que por sua vez tem como propósito descrever o tipo (atributos e relacionamento) de cada entidade de funcionário em um banco de dados qualquer. Normalmente, esse tipo de entidade pode vincular diversos subgrupos ou subtipos de suas entidades que expressam algum tipo de relevância e carecem de ser representados de maneira correta.

No exemplo apresentado pela Figura 3.12, será que você consegue identificar os tipos de entidade FUNCIONARIO existente? Vamos lá! Considere que a entidade do tipo funcionário é representada ora pelas entidades nomeadas de “secretária”, “engenheiro” e “técnico”. É possível interpretar que esse conjunto de entidades estão por sua vez vinculadas ao um conjunto de entidades “funcionário”, isso é, cada entidade é considerada também membro de qualquer um desses subtipos de funcionário.

Sendo assim, o tipo de entidade nomeada de FUNCIONARIO é considerado superclasse (genérica) ou supertipo de cada uma das subclasses (especializadas).

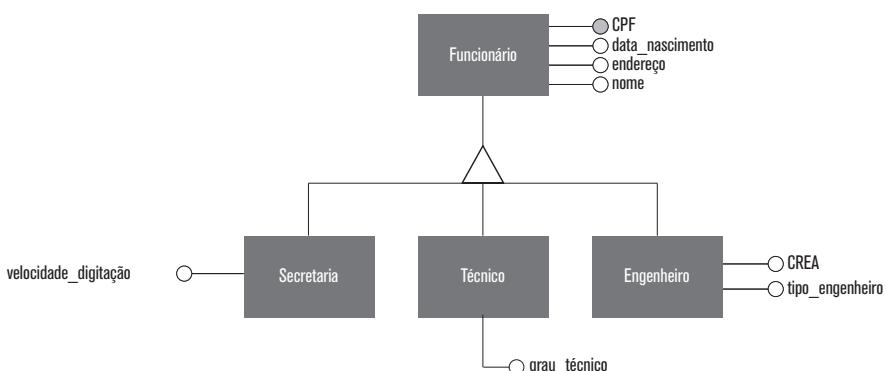


Figura 3.12 – Exemplo de entidade genérica e entidades especializadas

Dessa maneira, podemos considerar que especialização é um processo pelo qual é possível determinar um conjunto de subclasses de um tipo de entidade. Tal subconjunto de subclasses forma uma especialização tomando como referência as variadas características da superclasse, a citar como exemplo, secretária, engenheiro e técnico, ou seja, simplesmente se refere às especializações da superclasse FUNCIONARIO, que distingue as entidades de funcionário pelo uso do tipo de cargo.

Nesse contexto, o conjunto de entidades secretaria, inclui além do atributo específico velocidade_digitação, os atributos CPF, nome, data_nascimento e endereço, esses herdados da entidade FUNCIONARIOS.

Repare também que a uma instância de secretária, por exemplo, também é considerada como uma instância de funcionário, ou seja, um determinado membro da subclasse também se torna membro da superclasse, todavia, exercendo funcionalidades distintas.

Outro quesito relevante refere-se a possibilidade de existir várias especializações do mesmo tipo de entidade, considerando apenas as propriedades particulares, a citar como exemplo, outra especialização vinculada a entidade funcionário, que poderia refletir na criação de duas novas subclasses, essas nomeadas respectivamente de FUNCIONARIO_MENSAL e FUNCIONARIO_HORISTA, onde, evidentemente, a forma de realizar o pagamento irá diferenciar os tipos de funcionários.

Em uma extensão do Modelo ER, se cada entidade do conjunto de entidade genérica tiver que aparecer obrigatoriamente em um dos subconjuntos de entidade especializada, considera-se que a especialização/generalização sendo como TOTAL.

Assumindo uma característica oposta, uma especialização/generalização é dita como PARCIAL quando uma entidade do conjunto de entidade genérica não possuir a obrigatoriedade de aparecer como uma entidade de um dos subconjuntos de entidade especializada.

Graficamente, o DER representa uma especialização/generalização TOTAL incluindo simplesmente a letra “t” em minúsculo do lado superior direito do triângulo utilizado para especificar as entidades especializadas. Entretanto, a representação de uma especialização/generalização PARCIAL é dada pelo uso da letra “p”, também em minúsculo, do lado superior direito do triângulo.

Para exemplificar o uso de uma especialização/generalização considerada TOTAL, visualize a Figura 3.13 onde um determinado funcionário poderá ser exclusivamente, secretária, técnico e ou engenheiro. Nesse exemplo, não considere que um funcionário não seja pelo menos uma secretária, um técnico e um engenheiro. Esse detalhe referente às possíveis especializações dos funcionários a serem aplicadas no projeto de banco de dados é reportada no ato da entrevista. Ainda assim, é possível nos depararmos com a possibilidade do projetista de dados especificar que um conjunto de entidade genérica deverá ser representada em mais de um conjunto de entidades especializadas.

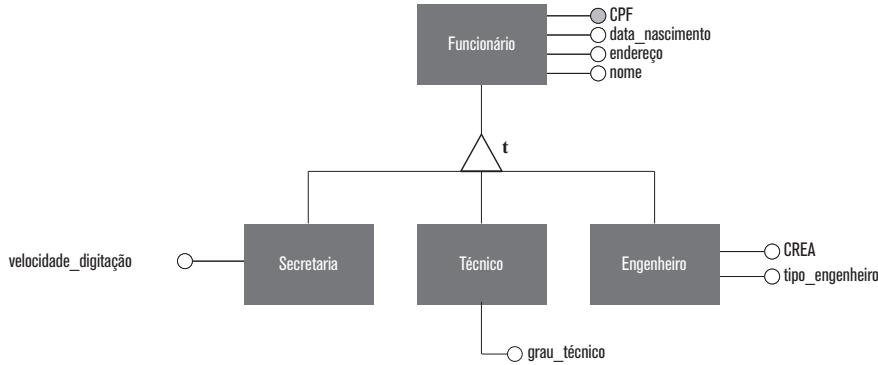


Figura 3.13 – Especialização/generalização TOTAL (t)

Uma especialização/generalização é considerada como sendo **EXCLUSIVA** quando cada entidade do conjunto de entidade genérica apresentar-se indispensavelmente no máximo em uma entidade do conjunto de entidade especializada. O oposto de especialização/generalização EXCLUSIVA é dito pela possibilidade de uma entidade do conjunto de entidade genérica apresentar-se como uma entidade em mais de um dos conjuntos de entidade especializada. Esse tipo de especialização/generalização é denominado de **COMPARTILHADA**.

A fim de representar um exemplo de especialização/generalização dita como EXCLUSIVA, graficamente o DER utiliza a letra “e” em minúsculo no lado superior do triângulo. Todavia, para representar um tipo de especialização/generalização COMPARTILHADA, também por meio do uso de um DER, simplesmente adicionamos a letra “c”, também em minúsculo no lado superior direito do triângulo.

É possível ainda, a existência de cenários que permite o uso simultâneo de diversos tipos de especialização/generalização, por exemplo, EXCLUSIVA e TOTAL ou EXCLUSIVA e PARCIAL, bem como, COMPARTILHADA e TOTAL ou COMPARTIPLHADA e PARCIAL. Entretanto, em nenhuma circunstância será permitido o uso de especialização/generalização que paralelamente seja COMPARTILHADA e EXCLUSIVA ou TOTAL e PARCIAL.

Para exemplificação considere o MER representado pela Figura 3.14, verifique que não existe nenhuma informação sobre a possibilidade de um determinado técnico também ser um engenheiro. Isso nos permite concluir que esse exemplo é de uma generalização/especialização COMPARTILHADA.

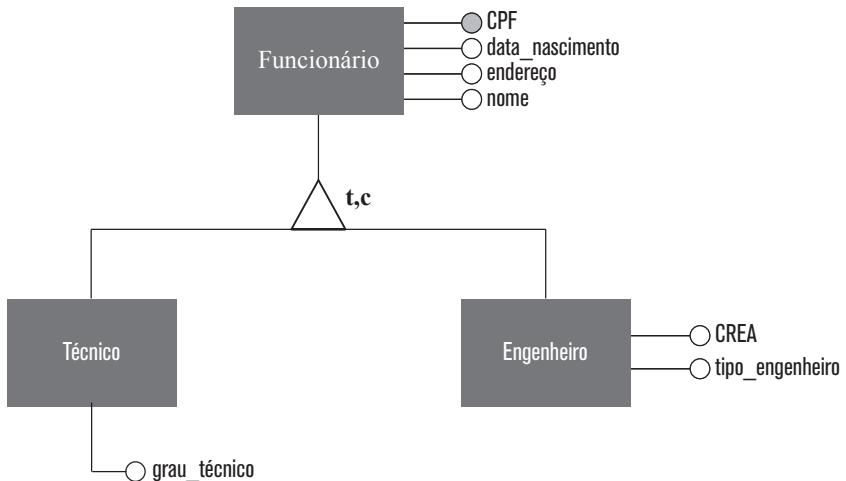


Figura 3.14 – Especialização/generalização do tipo COMPARTILHADA

Finalizando, conclui-se que todo esse processo de especialização permite que seja possível definir um conjunto de subclasses de um determinado tipo de entidade, e ainda, incluir atributos especializados para as subclasses como também especificar tipos de relacionamentos considerados exclusivos entre cada subclass e demais tipos de entidade ou ainda, outras subclasses.

3.3.2 Entidade Genérica

Uma entidade genérica é caracterizada pelo processo inverso de abstração o qual excluímos as diferenças encontradas entre os diversos tipos de entidade. Nesse processo, o objetivo é identificar adequadamente as características consideradas comuns, isso é, generalizar em uma exclusiva superclasse.

Entendeu o conceito de entidade genérica? Está confuso?! Não se preocupe, apresentaremos um exemplo para desmistificar o conceito de generalização.

Suponha a existência de dois tipos de entidade, uma por sua vez, identificada como CARRO e outra, identificada como CAMINHÃO, ambas visualizadas pela Figura 3.15. Note que existem diversos atributos considerados comuns entre as duas entidades, possibilitando que os mesmos sejam generalizados na entidade VEÍCULO. Dessa maneira, tanto CARRO quanto CAMINHÃO são considerados subclasses da superclasse generalizada VEÍCULO.

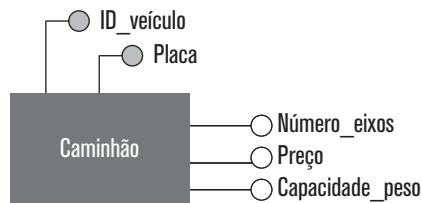
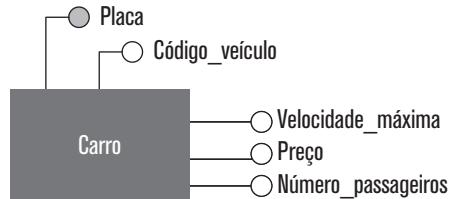


Figura 3.15 – Entidades especializadas CARRO e CAMINHÃO

Repare que foi realizado o processo oposto do processo de especialização, cujo objetivo foi estabelecer a generalização. Visualize por intermédio da Figura 3.16 que CARRO e CAMINHÃO tornam-se especializações da entidade genérica VEÍCULO.

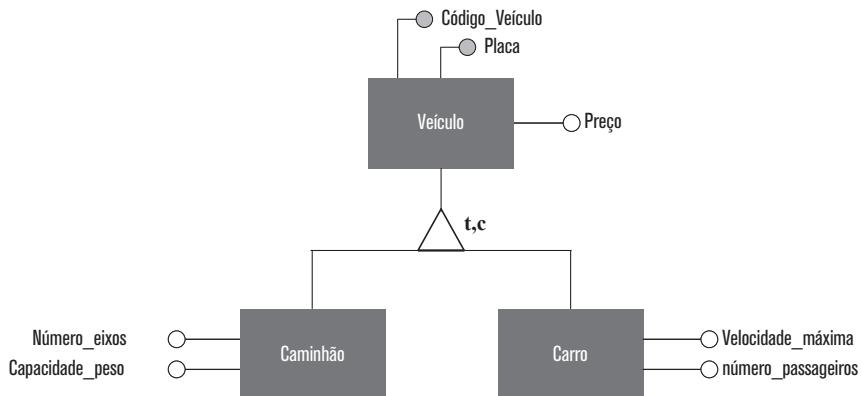


Figura 3.16 – Generalização (Veículo)

Concluímos mais um tópico importante da disciplina de Modelagem de Dados, explorando corretamente o recurso de herança, que permite que uma subclasse herde as propriedades consideradas comuns da superclasse.

3.3.3 Entidade Associativa

Na elaboração de um projeto de banco de dados, a confecção de um diagrama entidade-relacionamento (DER) exige que seja realizado eventuais descobertas, essas descobertas normalmente envolvem alguns tipos de entidades e seus respectivos relacionamentos. Inicialmente, o projetista de banco de dados elabora uma versão preliminar do projeto de banco de dados, e, com certeza, essa versão preliminar receberá novas sugestões/alterações a fim de atender/lapidar ainda mais os requisitos do negócio o qual se deseja armazenar os dados. Normalmente, em uma versão final, tem-se um número considerável de entidades e relacionamentos, deixando o DER na maioria das vezes indecifrável. Para essas ocasiões, é possível que se realize o agrupamento de entidades para tentar minimizar o número de entidades apresentadas no DER.

Essa associação entre entidades também é caracterizado como um tipo de entidade “virtual”, a qual é utilizada para simbolizar várias entidades e relacionamentos no DER. Uma entidade associativa é considerada “virtual” ou “abstrata” pelo simples fato de não constituir efetivamente uma entidade final válida no DER.

Já aprendemos em conceitos anteriores que, normalmente, em algumas situações, torna-se imprescindível associarmos uma entidade com a ocorrência de um relacionamento. É importante você recapitular que o MER não possibilita em nenhuma circunstância que seja realizado associações entre relacionamentos, apenas entre entidades. Sendo assim, uma entidade associativa tem como propósito vincular um relacionamento como se o mesmo fosse uma entidade, conforme apresentado como exemplo pela Figura 3.17.



Figura 3.17 – Agrupamento entre entidades (agregação)

Caso, eventualmente, fosse necessário controlar os medicamentos prescritos por um determinado médico após a realização de uma consulta, seria necessário vincular a entidade MEDICAMENTO com uma ocorrência de uma consulta, associando a entidade MEDICAMENTO com o relacionamento CONSULTA. É provável que você esteja achando isso tudo muito estranho. De fato você tem razão, pois essa manobra não é permitida. Não se deve realizar esse vínculo dessa maneira, a fim de solucionar o problema exposto, é aconselhado que o relacionamento CONSULTA torne-se uma entidade associativa, ora representada graficamente através de um retângulo envolto do relacionamento. A Figura 3.18 apresenta um exemplo do uso de uma entidade-associativa que envolveu por sua vez as entidades MÉDICO e PACIENTE, essas intituladas a partir do processo associativo de CONSULTA.

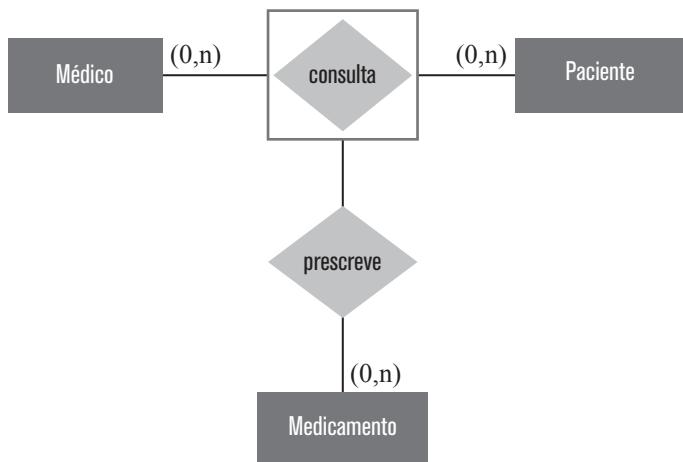


Figura 3.18 – Agrupamento das entidades “Médico” e “Paciente” resultando na entidade associativa “Consulta”

Esse processo também é denominado de agregação, isso é, um conjunto de relacionamentos e suas respectivas entidades são agregadas em uma nova entidade. Dessa forma, o exemplo exposto pela Figura 3.18 agrupa o relacionamento CONSULTA juntamente com as entidades MÉDICO e PACIENTE, constituindo uma nova entidade ora chamada de CONSULTA. Em algumas ocasiões, não é necessário estabelecer um nome para a nova entidade criada após o processo de agregação.

Na próxima Figura 3.19 é possível visualizar um exemplo de agregação aplicada no DER ora representado graficamente por um retângulo que por sua vez engloba as entidades e o relacionamento envolvido no processo. Ainda nos é permitido aplicar cardinalidades mínima e máxima no conjunto de relacionamentos constituintes da agregação. Para exemplificar, considere que uma determinada consulta pode ou não ter a prescrição de medicamentos.

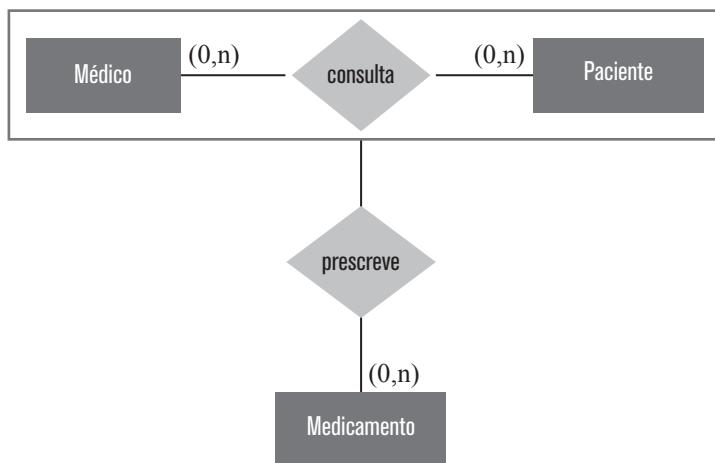


Figura 3.19 – Entidade associativa

3.4 Diagrama Entidade-Relacionamento (DER)

Antes de iniciarmos esse novo tópico, considere o diagrama entidade-relacionamento de um banco de dados de uma empresa imaginária, ora representado pela Figura 3.12 abaixo:

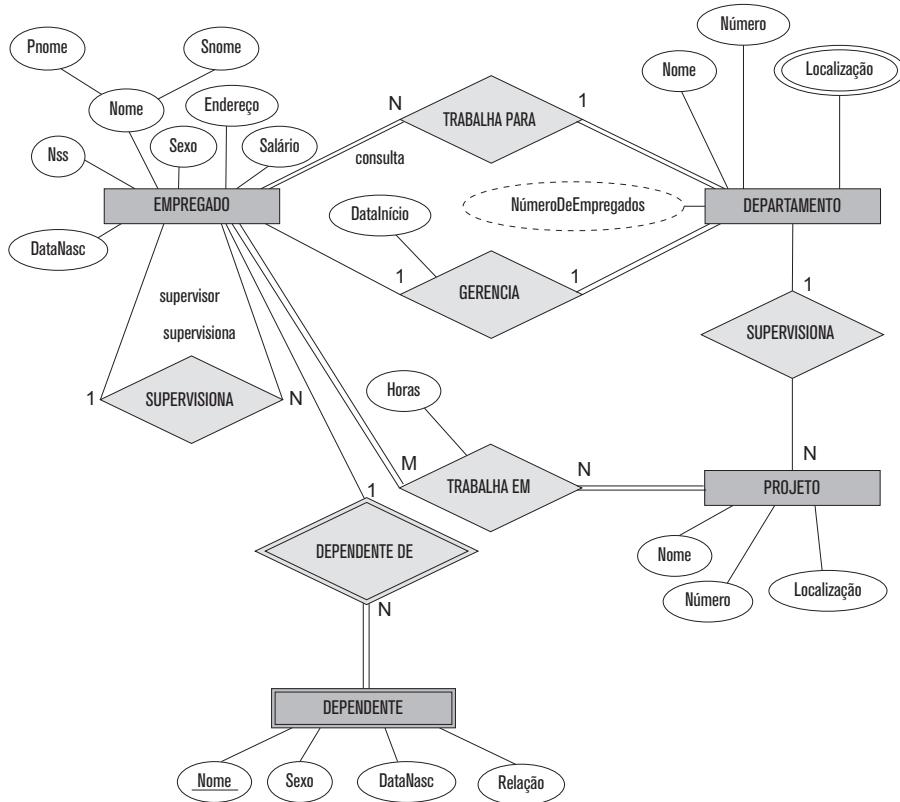


Figura 3.12 – Exemplo de Diagrama Entidade-Relacionamento

Com o conhecimento exposto até o momento, você certamente já identificou as principais entidades, essas intituladas de **EMPREGADO**, **DEPARTAMENTO** e **PROJETO**, que por sua vez, são representadas graficamente pelo uso de retângulos. Já os relacionamentos nomeados de **TRABALHA_PARA**, **GERENCIA**, **CONTROLA** e **TRABALHA_EM** utilizam a representação gráfica através do uso de losangos. Esses relacionamentos têm como intuito realizar a conexão das diversas

entidades participantes do modelo de dados ora proposto, a fim de atender as necessidades de uma regra de negócio específica. Os atributos (propriedades das entidades) são representados graficamente por meio do uso de elipses conectadas as entidades e ou, eventualmente, vinculadas aos relacionamentos. Existe ainda uma forma de representar atributos compostos, onde esses também são representados graficamente pelo uso de elipses, todavia, são associados aos atributos o qual depende, por exemplo, o atributo NOME é composto pelos atributos PNOME (primeiro nome) e SNOME (sobrenome). Outro detalhe que podemos abstrair diz respeito aos atributos multivalorados. Sua denotação é dada por meio de elipses circundada por linhas duplas, para exemplificar, repare o atributo LOCALIZAÇÃO da entidade DEPARTAMENTO. Os atributos identificadores (atributos-chaves e ou chave-primária) são identificados por sua vez na forma sublinhada, dentro da elipse. Como o próprio nome sugere, os atributos derivados são simbolizados pelo uso de elipse circundada de linhas tracejadas, a citar como exemplo, considere o atributo ora intitulado de NúmeroDeEmpregado pertencente a entidade DEPARTAMENTO. Em algumas literaturas você pode encontrar o conceito de que esse tipo de atributo (derivado) é também chamado de atributo processado, isso é, o valor atrelado a esse tipo de atributo estará sempre associado a algum tipo de processamento computacional.

Já a identificação de entidades e relacionamentos considerados fracos é imposta pelo uso de retângulos e losangos circundados com linhas duplas. Em nosso exemplo (Figura 3.12), podemos destacar esse conceito, ora representado pela entidade-fraca, identificada de DEPENDENTE e seu respectivo relacionamento, esse nomeado de DEPENDENTE_DE.

Esmiuçando ainda mais nosso exemplo de DER, apresentado pela Figura 3.12, é possível obter informações acerca das cardinalidades (multiplicidade) aplicada para cada relacionamento de grau 2 (relacionamento binário). Como exemplo, podemos citar a cardinalidade 1:1 existente entre o relacionamento GERENCIA que associa as entidades DEPARTAMENTO e EMPREGADO. A interpretação dessa cardinalidade seria: um empregado gerencia no máximo um departamento e, por sua vez, um departamento é gerenciado por no máximo um empregado. Outra cardinalidade digna de exemplo é a cardinalidade referente ao relacionamento TRABALHA_PARA que vincula as entidades DEPARTAMENTO e EMPREGADO, essa cardinalidade é dita como 1:N (“um” para “muitos”) e M:N (“muitos” para “muitos”) para o relacionamento TRABALHA_EM. Com relação a representação gráfica utilizada para sinalizar uma restrição de parti-

cipação parcial, utiliza-se o emprego de linhas simples, todavia, quando necessitamos de representar graficamente no DER uma dependência existente entre entidades, utilizamos linhas paralelas.

Outro detalhe considerado relevante que devemos prestar muita atenção, discorre sobre a existência de um auto-relacionamento. Vimos anteriormente que, quando existe um auto-relacionamento em um DER qualquer, é fundamental o uso de papéis a fim de identificar adequadamente qual é o tipo de relacionamento desempenhado em um determinado instante. Como exemplo, considere ainda o nosso DER da Figura 3.12, o auto-relacionamento nomeado de SUPERVISIONA assume o uso de dois papéis díspares, ou seja, ora o empregado assume o papel de SUPERVISOR ora o papel de SUPERVISIONADO.

3.4.1 Grau de Relacionamento

Para determinar o grau de um relacionamento devemos analisar o número de entidades participantes do mesmo relacionamento. Dessa maneira, é possível identificar como grau um (também chamado de unário) um relacionamento que utiliza apenas uma entidade, por outro lado, um relacionamento de grau dois (binário) é aquele que faz uso de duas entidades, grau três (ternário) utiliza três entidades e grau “n” (“n”ário) faz uso de mais de três entidades. Em nosso próximo DER, ora representado pela Figura 3.13, podemos visualizar um relacionamento ternário.

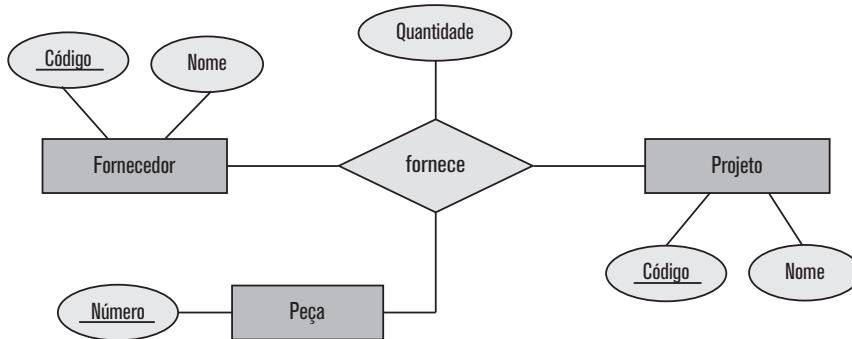


Figura 3.13 – Relacionamento ternário (grau três)

A Figura 3.14 nos demonstra um DER a fim de exemplificar um relacionamento binário. Repare que o relacionamento intitulado de CURSO emprega apenas dias entidades (ALUNO e DISCIPLINA).

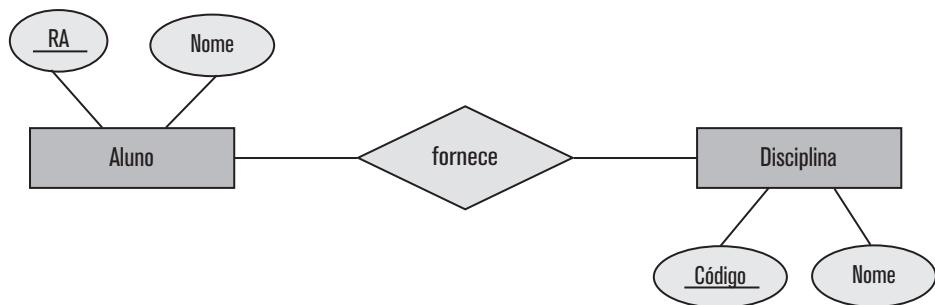


Figura 3.14 – Relacionamento binário (grau dois)

A Figura 3.15 nos apresenta outro exemplo, onde é possível identificarmos que o grau do relacionamento CASAMENTO é unário (grau um), simplesmente por existir uma única entidade, essa chamada de PESSOA. Você não deve se esquecer de que, quando fazemos uso de um relacionamento unário, é imprescindível o emprego de papéis. Nesse tipo de relacionamento, os papéis possuem um importante objetivo, o qual é representar corretamente uma associação entre ocorrências de uma mesma entidade (isso é, ora pessoa assume o papel “homem” ora pessoa assume o papel “mulher”).

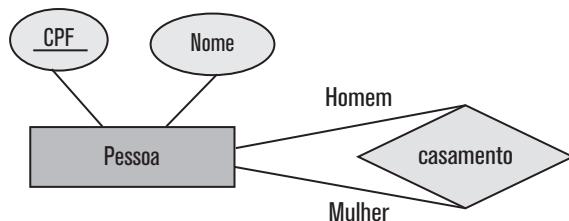


Figura 3.15 – Relacionamento unário (grau um)

Finalizando o tópico que discorre sobre o grau dos relacionamentos, é possível visualizarmos por meio da Figura 3.16, um exemplo de relacionamento “n”ário, ou seja, é possível abstrair uma associação entre “n” (diversas) ocorrências de entidades.

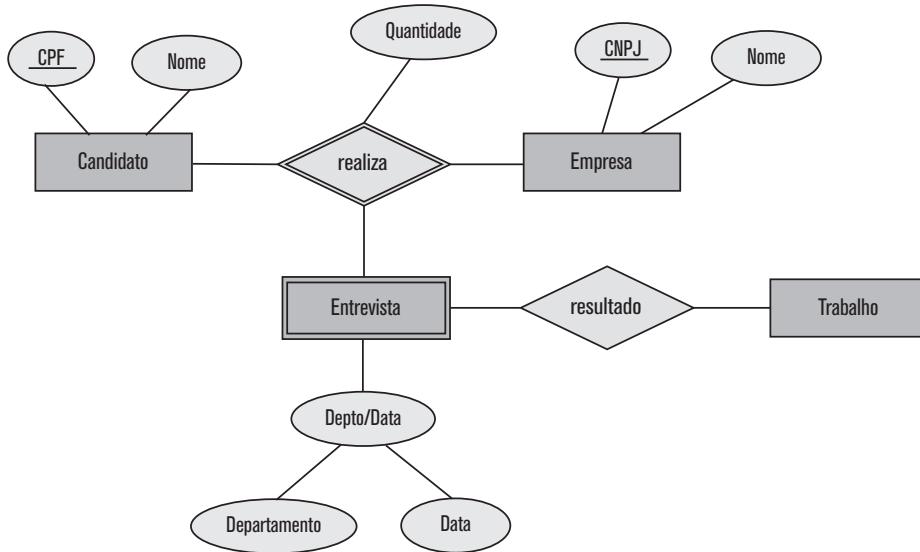
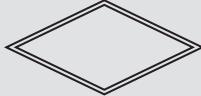
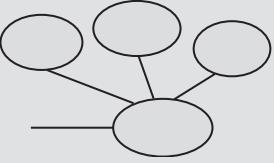


Figura 3.16 – Exemplo de relacionamento “n”ário (várias entidades)

Você deve ter percebido as variadas notações utilizadas para representar uma entidade, entidade-fraca, relacionamento, atributo, atributo identificador, etc., em um diagrama entidade-relacionamento (DER). A Tabela 3.1 a seguir apresenta as diversas notações com suas respectivas descrições, empregadas até o presente momento nos exemplos de DER. Lembrando que, eventualmente, essas representações gráficas (símbolos) podem sofrer variação dependendo o tipo de ferramenta utilizada na elaboração dos diagramas entidade-relacionamento.

NOTAÇÃO (SIMBOLOGIA)	DESCRIÇÃO
	Entidade
	Entidade-Fraca
	Relacionamento
	Relacionamento Identificador
	Atributo
	Atributo-chave (identificador)
	Atributo Multivalorado

NOTAÇÃO (SIMBOLOGIA)	DESCRIÇÃO
	Atributo Composto
	Atributo Derivado

3.5 Modelando o “negócio”

Bem, agora que você já aprendeu basicamente todos os conceitos sobre o diagrama entidade-relacionamento, chegou o momento de coloca-los em prática.

A descrição dos requisitos a seguir, permite que utilizemos uma entidade-fraca. Sendo assim, vamos elaborar a partir desses requisitos o DER que completa em sua totalidade, as regras do negócio, ora representado por uma universidade fictícia.

Uma universidade deseja implantar um Sistema de Informação para realizar o gerenciamento da quantidade de carteiras por sala de aulas existentes nos seus vários Campus. Hoje, essa universidade possui 5 campus, localizados em regiões (cidades) distintas. Em cada campus existe blocos de salas. Cada campus possui um número sequencial utilizado para identificação e cada bloco de salas, por sua vez, é identificado por uma letra. As salas são numeradas de forma sequencial dentro de cada bloco. É desejável que o sistema de informação disponibilize algumas informações, como por exemplo: nome e endereço de cada um dos Campus; para cada campus, quais são os blocos de sala de aula e para cada bloco qual é o número de salas que possui, juntamente com a quantidade de andares; para cada sala de aula, é importante conhecer o número de carteiras, seu tamanho (área) e quais são as carteiras existente em cada sala (número do patrimônio da carteira e se o braço de apoio está localizado do lado direito ou esquerdo).

Após abstrairmos os requisitos necessários, podemos confeccionar uma versão preliminar do DER, o qual deve atender de forma adequada todas as fun-

cionalidades expostas na descrição do Sistema de Informação a ser criado pela Universidade. A Figura 3.17 apresenta um exemplo de DER que teve como propósito atender os requisitos triviais do Sistema de Informação da Universidade. Você pode identificar a utilização de duas entidades-fracas, uma chamada de “bloco” e outra de “sala de aula”. Visualize os detalhes pertinentes a essas entidades-fracas no box cinza.

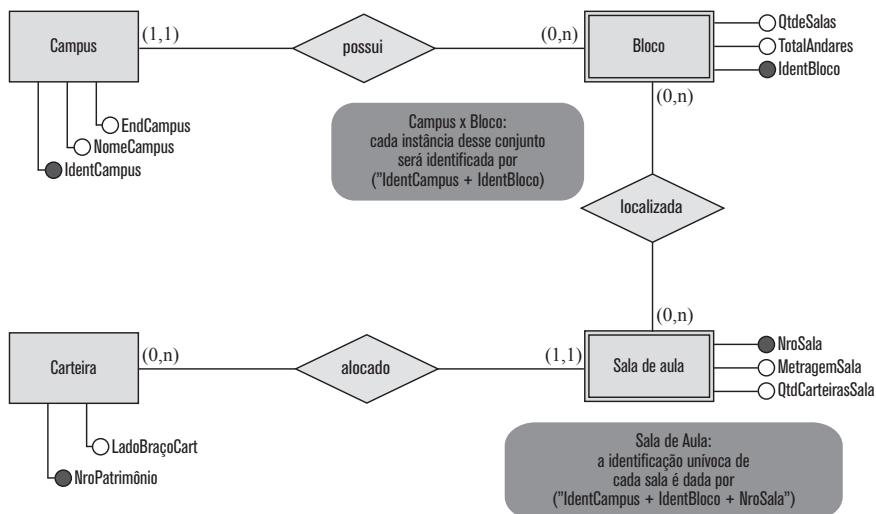


Figura 3.17 – DER de uma Universidade Fictícia



ATIVIDADE

9. Conceitue adequadamente um atributo, e discorra sobre os seus principais tipos. Na sequência, dê pelo menos um exemplo de cada tipo.
10. Conceitue um relacionamento e classifique os relacionamentos em relação ao número de objetos envolvidos.
11. Imagine um contexto acadêmico, o qual, poderíamos considerar uma sala de aula. Qual seria a cardinalidade máxima de um professor em relação aos alunos, como também, dos alunos em relação ao professor?



REFLEXÃO

Parabéns! Você finalizou mais um capítulo, essa sem dúvida foi uma densa unidade de estudos! Foi possível aprendermos acerca das principais características do Modelo Entidade-Relacionamento, e como esse modelo se torna imprescindível para a criação de um bom projeto de banco de dados.

Acreditamos que você tenha maximizado seu conhecimento referente ao MER, como seus principais componentes, a citar: tipos de entidades, relacionamentos, cardinalidade (um-para-um; um-para-muitos e muitos-para-muitos), tipos de atributos (simples, composto, multivariado, derivado e identificador).

Foi possível ainda estudarmos acerca da representação gráfica do MER, essa imposta pelo uso do que chamamos de diagrama entidade-relacionamento (DER), sobretudo interpretarmos corretamente o grau dos relacionamentos existentes nele.

Você consegue lembrar-se dos principais graus que podemos abstrair de um determinado relacionamento?

O grau de um relacionamento é factível de ser abstraído levando em consideração o número de entidades participantes (únário, binário, ternário, quaternário e "n"ário).



LEITURA

Artigos on-line: para você aumentar ainda mais o seu nível de conhecimento sobre os MER e DER:

<http://www.devmedia.com.br/projeto-de-bd-tatico-para-informacoes-da-concorrencia/31392>
<http://imasters.com.br/artigo/8568/banco-de-dados/documentacao-de-projetos-web-der/>

Livros:

Modelagem Lógica de Dados: construção básica e simplificada, de Eduardo Bernardes Castro.
Banco de Dados: Implementação em SQL, PL/SQL e Oracle 11g, Sandra Puga, Edson França e Milton Goyga;

Nestes livros você encontra um bom conteúdo para complementar os estudos apresentados pela apostila. Aprofunde seus conhecimentos!



REFERÊNCIAS BIBLIOGRÁFICAS

ELMASRI, R.; NAVATHE, S.B. Sistemas de bancos de dados. São Paulo: Pearson (Addison Wesley), 2005.

KORTH, H.; SILBERCHATZ, A. Sistemas de bancos de dados. 3. ed. São Paulo: Makron Books, 1998.

HEUSER, C. A. Projeto de Bancos de Dados. 2. ed. Porto Alegre: Sagra Luzzatto, 1999.

PRESSMAN, R.S. Engenharia de software. São Paulo: Makron Books, 1995.

ROB, P; CORONEL, C. Sistemas de Banco de Dados: Projeto, Implementação e Administração. 8. Ed. São Paulo: Cengage Learning, 2011.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: Elsevier, 2006.

DATE, C. J.; Introdução a Sistemas de Banco de Dados; 8^a. Ed.; Trad. Daniel Vieira; Rio de Janeiro: Elsevier, 2003.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4 ed. Instituto de Informática da UFRGS, Sagra DC Luzzatto, 1998.

RAMAKRISHNAN, R. GEHRKE, J. Database Management Systems. 2. ed. Boston: McGraw-Hill, 2000.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T.; JAGADISH, H. V.; Database Modeling and Design – Logical Design. 5^a ed., Burlington – USA: Elsevier, 2011.

CASTRO, E. B.; Modelagem Lógica de Dados: construção básica e simplificada; Rio de Janeiro: Editora Ciência Moderna, 2012.

PUGA, S.; FRANÇA, E.; GOYA, M.; Banco de Dados – Implementação em SQL, PL/SQL e Oracle 11g; São Paulo: Pearson Education do Brasil, 2013.



NO PRÓXIMO CAPÍTULO

No próximo capítulo iremos esmiuçar o modelo de dados relacional! Pratique exaustivamente à modelagem de dados, os tipos de relacionamentos e os seus principais conceitos.

4

Modelo de Dados Relacional

4 Modelo de Dados Relacional

No capítulo anterior estudamos acerca dos modelos de dados, com sua evolução e a importância de compreendê-los para que seja possível a elaboração de um projeto de banco de dados bem estruturado.

Caso você tenha alguma dúvida sobre algum conceito, retorne a unidade anterior e realize uma revisão dos principais conceitos!

Estudamos profundamente sobre os principais modelos de dados, vamos a partir de agora dar ênfase no modelo relacional, o qual é considerado um modelo clássico, e amplamente adotado pela grande parte dos SGBDs. Estudaremos os conceitos e princípios relevantes, como os atributos identificadores, restrições de integridade, mapeamento do MER para o modelo relacional, dentre outros.

Não perca o foco neste conteúdo!



OBJETIVOS

Este capítulo tem como objetivo permitir um estudo detalhado referente aos aspectos pertinentes ao modelo de dados relacional, onde, seus principais conceitos podem ser encontrados nos principais SGBDs comerciais. Esperamos que ao final desta unidade você seja capaz de compreender o funcionamento do uso de chave primária e estrangeira, como as regras de integridade de entidade e referencial são imprescindíveis para promover a integridade dos dados, além de, estudarmos através de exemplos práticos a tarefa do mapeamento do MER para o modelo de dados relacional.



REFLEXÃO

Você se lembra dos conceitos estudados na unidade anterior?

Quanto aos principais graus aplicados aos relacionamentos?

Você se sente confortável em discorrer sobre os principais componentes de um DER?

Você se encontra confortável em distinguir adequadamente os conceitos acerca dos Modelos de Dados: Hierárquico, em Rede e Relacional?

4.1 Modelo De Dados Relacional

O Modelo de Dados Relacional foi criado por Codd na década de 70. Esse modelo de dados é caracterizado por ser o mais simples dos modelos de dados disponíveis para implementação de banco de dados. Tal modelo possui como objetivo a apresentação dos dados similar a um conjunto de relações. Dessa maneira, podemos comparar uma relação como sendo uma possível tabela, ou, simplesmente, um simples arquivo contendo “n” registros.

O Modelo de Dados Relacional é calcado no conceito de matrizes. Podemos considerar que as linhas em uma matriz como sendo os registros e as colunas, seus respectivos campos. Os identificadores das tabelas (relação) e dos campos são de extrema relevância para seu entendimento entre o que você está armazenando, onde está armazenando e qual a relação existente entre os dados armazenados.

Como exemplo, tomamos a Figura 4.1, que estrutura os dados por meio do uso de um modelo de dados relacional.

	RA	Nome	Sala	Departamento
tb_Aluno	192	Ana Paula	1	Computação
	324	Cecília	2	Computação

	Código	Nome	Créditos	Departamento
tb_Disiplina	C123	ICC	4	Computação
	C342	Estrutura de Dados	4	Computação
	M098	Cálculo I	4	Matemática
	C124	Banco de Dados	4	Computação

tb_pré_requisito	Código	Pré_requisito
	C124	C324
	C124	M098
	C324	C123

tb_seção	Código	Disciplina	Semestre	Ano	Professor
	45	C123	1	2012	Prof. Ms. Frederico Valeri
	52	C324	2	2012	Prof. Dr. Rodrigo Mendes
	124	C124	1	2013	Prof. Ms. José Ribeiro
	132	M098	1	2012	Profa. Ms. Valéria Cruz
	139	C324	2	2013	Prof. Dr. Jean Nunes
	155	C124	1	2013	Prof. Ms. Givanilso Martins

	RA	Código_Seção	Nível
tb_histórico	192	155	B
	192	139	B
	324	132	A
	324	124	A
	324	52	A
	324	45	B

Figura 4.1 – Exemplo de um Modelo de Dados Relacional

As relações apresentadas pela Figura 4.1 podem ser vistas também como tabelas. Em uma tabela, uma linha representa um registro (tupla), isso é uma coleção de valores relacionados. Nesse contexto, esses valores referem-se a um fato de uma entidade específica, ou até mesmo uma instância de um relacionamento qualquer. Vimos no início do tópico que o nome da tabela com os nomes das respectivas colunas são cruciais para que seja possível interpretarmos o correto sentido dos valores em cada linha (tupla) existente em uma tabela. Bem, você conseguiu entender esse conceito? Não se preocupe! Vamos agora interpretar as relações pertinentes a Figura 4.1. É factível identificarmos que a tabela ora nomeada de tb_aluno, cada linha (tupla) diz respeito a um fato específico ora armazenado nessa tabela. Suas colunas, essas identificadas como RA (registro acadêmico), nome, sala e departamento possibilitam que realizemos a devida interpretação dos valores vinculados para cada linha (tupla). Considere que, para cada coluna, todos os valores associados a mesma, deverá, obrigatoriamente, manipular o mesmo tipo de dado.

Na maioria das literaturas que discorre sobre o tema Modelo Relacional, uma linha é caracterizada como sendo uma tupla, uma coluna como um atributo e uma tabela de relação. Ainda tomando como referência o modelo relacional, o conceito de domínio refere-se ao tipo de dados que será armazenado em uma coluna qualquer. O grau de uma relação é interpretado pelo número de atributos existentes nela.

A fim de esclarecer o grau de uma relação, considere a relação ora apresentada abaixo, que por sua vez, apresenta informações referentes aos alunos de uma universidade qualquer:

Aluno(*ra, nome, endereço, telefone, celular, idade, media_ponderada*).

Nessa relação, consideramos que aluno é o nome da relação, a qual possui sete atributos, por isso podemos interpretar que essa relação é de grau sete. É possível ainda, detalhar alguns domínios para cada atributo presente na relação aluno, por exemplo, o domínio RA (registro acadêmico – numérico ou literal), nome (nomes dos alunos - literal), telefone (número do telefone fixo – numérico ou literal), celular (número do telefone móvel – numérico ou literal) e media_ponderada (valor correspondente a média ponderada de um determinado aluno - numérico).

A Figura 4.2 nos apresenta a relação intitulada de aluno, onde podemos abstrair que cada tupla (linha) representa uma entidade aluno. Essa relação representada por sua vez como uma tabela considera cada tupla como sendo uma linha e, cada atributo existente nessa tabela, como um cabeçalho que, tem como propósito indicar um papel específico ou simplesmente promover a interpretação dos valores vinculados a cada coluna.

	RA	Nome	Endereço	Telefone	Celular	Idade	Média Ponderada
	123	Alice Martins	Rua Piauí, 87	3412-0909	null	21	6,5
	324	Marcia Garcia	Av. Pio XII, 1562	null	9123-0989	26	7,8
Aluno	709	Bruno Freitas	Rua Treze de Maio, 98	null	8345-0912	19	9,0
	21	Murilo Henrique	Rua Javari, 23	3634-0987	null	20	4,8
	523	Igor Afonso	Av. Nove de Julho, 23	null	8723-8877	23	7,0

Tuplas

Figura 4.2 – Tabela “aluno” com suas respectivas instâncias



ATENÇÃO

Um domínio descreve os tipos de dados que cada campo (atributo) pode armazenar. (Heuser, 2004)

4.2 Chave Primária (Atributo Identificador)

Com os conceitos estudados até o presente momento, já aprendemos que uma relação é considerada como um conjunto de tuplas. Devemos considerar que, todos os elementos constituintes desse conjunto de tuplas devem ser exclusivos, isto significa que, absolutamente nenhuma tupla deverá possuir a mesma combinação de valores para todas as suas colunas. Isso significa que, em uma relação, eventuais subconjuntos de atributos não podem conter tuplas com a mesma combinação de valores. Sendo assim, conclui-se que toda a relação deve conter pelo menos uma super-chave. Como exemplo, retornamos a Figura 4.2, onde o atributo nomeado de RA (registro acadêmico) é considerado uma super-chave da relação aluno por simplesmente não deixar que um mesmo aluno possua o mesmo registro acadêmico.



CONEXÃO

Como sugestão, aconselhamos a leitura deste artigo, para aprofundar seus conhecimentos no que diz respeito ao modelo relacional:

<http://imasters.com.br/artigo/2419/banco-de-dados/o-modelo-relacional-de-dados-parte-01/>

<http://imasters.com.br/artigo/5403/banco-de-dados/modelagem-de-dados-parte-05-transformacao-entre-modelos/>

Um valor vinculado a um atributo-chave é utilizado para distinguir de maneira exclusiva uma tupla em uma relação específica. Ou seja, o valor 21, do atributo RA, é utilizado exclusivamente para identificar a tupla vinculada ao aluno “Murilo Henrique”, ora existente na relação ALUNO. É importante escolher corretamente o atributo-chave de uma relação, principalmente se considerarmos que o mesmo não poderá sofrer modificações no transcorrer do tempo. A fim de melhor exemplificar, é possível dizer que o atributo NOME da mesma relação ALUNO

não deverá, em hipótese alguma, ser escolhido como atributo-chave, sobretudo por, não possuirmos garantia da não existência de nomes idênticos.

Na maioria das vezes, em uma relação poderá existir mais de um atributo-chave. Para atender essas particularidades, cada chave é referenciada como chave-candidata. Voltamos ao exemplo ora apresentado pela Figura 4.2, em algumas ocasiões, seria permitido adicionar um atributo chamado de **CODIGO_ALUNO** (*surrogate key*) para identificar exclusivamente um determinado aluno por meio de um código interno. Nessa circunstância, a relação ALUNO teria duas chaves-candidatas, o atributo RA e CODIGO_ALUNO. Esse recurso é considerado normal, isso é, é relativamente comum escolhermos duas chaves-candidatas como chave-primária (atributo-chave) de uma relação. No modelo de dados relacional, representamos uma chave-primária, simplesmente sublinhando os atributos que formam a chave-candidata. Se, eventualmente, uma relação possuir várias chaves-candidatas, a escolha da chave mais adequada se torna facultativa, porém, a seleção da chave-primária com o menor número de atributos é sempre desejável.



ATENÇÃO

Uma chave-candidata é uma chave que apresenta obrigatoriamente as duas características a seguir:

- 1) Unicidade: Não há duas linhas (tuplas) distintas na tabela com o mesmo valor para os atributos da chave (isso já era garantido na chave e foi explicado na seção anterior);
- 2) Irreduzibilidade: Não existe um subconjunto de atributos da chave que apresentem a característica de Unicidade.

(Date, 2004)

Uma chave-primária inclui algumas características relevantes, a considerar a seguir:

- Unívoca (exclusiva): os atributos definidos para constituir a chave-primária, por definição, têm que possuir valores únicos para cada registro (ou tupla) na relação. Isso significa que todas as linhas devem ser identificadas de maneira exclusiva por essa chave. Nesse contexto, mencionamos que a tabela apresenta integridade de entidade;
- Não nula: nenhum dos atributos que constituem a chave primária poderá, em hipótese alguma, possuir valores nulos em nenhum registro;

- Não redundante: no caso da chave-primária ser composta, não poderá ser adicionado mais atributos do que os mínimos necessários para identificar os registros de forma unívoca.

Reflita: O que você entende sobre redundância controlada?

A redundância controlada é considerada um dos princípios do funcionamento dos bancos de dados relacionais, permitindo que distintas tabelas compartilhem um mesmo atributo.

4.3 Restrições de Integridade

Em um esquema relacional, S é formado por um conjunto de relações $S=\{R_1, R_2, \dots, R_n\}$, e, concomitantemente, por um conjunto de restrições de integridade (RI). Normalmente, atendemos duas consistências, essas consideradas cruciais, a citar Restrição de Integridade de Entidade (RIE) e a Restrição de Integridade Referencial (RIR). A RIE tem como propósito garantir o acesso aos dados sem nenhuma ambiguidade. Para exemplificar o emprego da RIE, considere uma tupla qualquer, ora existente na relação R, dizemos que o valor de cada atributo que constitui a chave-primária de (t) deve ser diferente de nulo e, ainda, não poderá haver uma outra tupla (t) em R com o mesmo valor da chave-primária de (t).

A fim de exemplificar a RIR, imagine uma tupla (t) e uma chave-estrangeira (foreign key) em (t), o valor da chave-estrangeira pode ser nulo se e somente se os atributos de chave-estrangeira não formarem a chave-primária de (t), e ainda, o valor da chave-estrangeira poderá ser diferente de nulo apenas se existir uma tupla (t) na relação referenciada tal que a chave-primária de (t) possuir o mesmo valor da chave-estrangeira de (t).

Existem ainda, outras implicações que devemos nos atentar referente a RIR, por exemplo, suponha a existência de duas relações, essas identificadas de R_1 e R_2 , respectivamente e uma chave-estrangeira (fk) aplicada à R_1 que referencia por sua vez à chave-primária de R_2 , três ações básicas podem ser consideradas:

- XII. Impedimento: uma operação de atualização não será concluída;
- XIII. Cascata: se ocorrer uma exclusão de uma tupla (t) qualquer ora armazenada em R_2 , então deverá ser removida todas as tuplas de R_1 tal que a chave-estrangeira referencie a chave-primária de (t); se ocorrer uma alteração da chave-primária de uma tupla (t) de R_2 , deverá então ser alterado todas as tuplas de R_1 que faça referência ao valor anterior da chave-primária de (t) para o novo valor da chave-primária de (t);

XIV. Anulação: se ocorrer uma exclusão ou uma alteração de uma tupla (t) de R_2 , dessa maneira para toda tupla de R_1 tal que a chave-estrangeira refencie a chave-primária de (t) deverá configurar o valor da chave-estrangeira para nulo.

Na Figura 4.3, podemos visualizar um esquema de banco de dados relacional o qual atende os requisitos básicos de uma empresa fictícia qualquer. O conceito de banco de dados relacional discorre de um esquema e suas respectivas instâncias.

Empregado												
<u>Cód.</u> <u>Emp</u>	Nome	Sobre- nome	Data- Nasc	Endere- ço	Sexo	Sa- lário	Cód. Dep- to					
Departamento												
<u>Cód.Depto</u>	Nome		<u>Cód.Gerente</u>		DataInício Gerente							
Dept_Localização												
<u>Cód.Depto</u>		<u>Dept.Localização</u>										
Projeto												
<u>CódProjeto</u>	Nome		ProjLocalização		<u>Cód.Depto</u>							
Trabalho_Projeto												
<u>Cód.Empregado</u>	<u>Cód.Projeto</u>		Horas									

Dependente					
<u>Cód Empregado</u>	<u>Cód Depen- dente</u>	Nome	Sexo	Data- Nasc	Relação

Figura 4.3 – Esquema Relacional de uma organização fictícia

Analizando o esquema relacional apresentado pela Figura 4.3, é possível identificar que o atributo chamado CódDept da relação intitulada de DEPARTAMENTO e DEPTO_LOCALIZAÇÃO diz respeito a um identificador aplicado ao departamento, atendendo o mesmo conceito aplicado em um cenário real. Dessa forma, esse mesmo conceito pode ser aplicado ao atributo intitulado de CódDept presente nas relações EMPREGADO e PROJETO. Para o nosso exemplo, utilizamos nomes iguais, todavia, isso não é regra, podemos utilizar nomes diferentes para os mesmos atributos, sem qualquer tipo de restrição. Normalmente, atributos que simbolizam conceitos dispareys podem utilizar o mesmo nome desde que estejam presentes em relações distintas. A caráter de exemplificação, o atributo ora identificado de NOME presente na relação EMPREGADO, por sua vez, utiliza o mesmo identificador, NOME, na relação DEPARTAMENTO.

Eventuais restrições de integridade poderão ser utilizadas no esquema de banco de dados relacional apresentado pela Figura 4.3, como, por exemplo, as restrições de chave-primária que tem como objetivo especificar as chaves-candidatas de cada relação em nosso esquema. Isso significa que, os valores atrelados às chaves-candidatas deverão garantir exclusividade para todas as tuplas. Outros dois tipos de restrições poderão ser aplicadas em nosso modelo relacional, a restrição de integridade de entidade (RIE) e a restrição de integridade referencial (RIR).

A restrição de integridade de entidade (RIE) que tem como objetivo determinar que absolutamente nenhum valor vinculado à chave-primária seja nulo, simplesmente por utilizamos esse valor de chave-primária para identificar unicamente as tuplas na relação. Como exemplo, assuma que existam duas tuplas ou mais com valores nulos associados à chave-primária. Dessa forma, não existiria nenhuma maneira de identificar exclusivamente uma tupla da outra.

A outra restrição chamada de restrição de integridade referencial (RIR) possui como propósito contemplar uma restrição associada entre duas relações, que garante a consistência entre tuplas das mesmas. Ou seja, a RIR garante que uma tupla de uma relação, essa associada por sua vez à outra relação, deverá,

referenciar a uma tupla existente naquela mesma relação. A Figura 4.4, é fácil observar que o atributo chamado de CódDeptº da relação EMPREGADO referencia um número de departamento em que cada empregado realiza suas atividades. Ou seja, todos os valores associados à coluna CódDeptº da relação EMPREGADO deverão pertencer ao conjunto de valores da coluna CódDeptº da relação DEPARTAMENTO.

As restrições de integridade referencial são provenientes dos relacionamentos entre conjuntos de entidades que por sua vez, representam um determinado esquema de banco de dados.

Outras regras de integridade podem ser aplicadas no modelo relacional, a citar as restrições não-nula (NOT NULL) e não-redundante (UNIQUE). A restrição não-nula poderá ser aplicada em uma coluna para garantir que todas as tuplas da tabela apresentem um valor para essa coluna. A restrição não-redundante é implementada em uma coluna para garantir que não exista nenhum valor duplicado na mesma. (DATE, 2003)

Empregado							
Cód. Emp	Nome	Sobrenome	Data- Nasc	Ende- reço	Sexo	Salário	Cód- Deptº
101	Joaquim	Mendes	09/01 /1995	Rua A, 1	M	R\$3.500,00	15
102	Elton	da Cruz	08/12 /1965	Rua B, 234	M	R\$2.350,00	15
103	Juliano	Batista	19/07 1965	Av C, 78	M	R\$1.775,00	24
104	Vanda	da Silva	30/06 1961	Rua D, 98	F	R\$4.565,00	24
105	Emersom	Frota	16/09 /1968	Rua E, 21	M	R\$9.100,00	15
106	Fabrício	dos Santos	31/07 /1986	Av J, 1347	M	R\$8.100,00	15

Empregado							
107	Michelle	Pavereli	30/03 /1989	Av H, 76	F	R\$2.450,00	24
108	Venilton	José	02/11 /1965	Rua T, 76	M	R\$8.899,00	11

Departamento			
<u>Cód Dep- to</u>	Nome	CódGe- rente	Datal Inico Gerente
11	Recursos Humanos	105	20/05/2000
15	Contabilidade	108	13/01/1996
24	Técnologia da Informação	106	13/02/2009

Deptos_Localização	
<u>CódDeptos</u>	<u>DeptosLocalização</u>
11	Sertãozinho
15	Mococa
24	RibeirãoPreto

Projeto			
<u>CódProjeto</u>	Nome	ProjLocalização	CódDeptos
11	Matriz	Sertãozinho	11
12	Alpha	Sertãozinho	11
13	Genexus	RibeirãoPreto	24

20	IRPF 2013	Mococa	15
30	ERP	RibeirãoPreto	24
40	SAP	RibeirãoPreto	24

Trabalho_Projeto		
<u>CódEmpregado</u>	<u>CódProjeto</u>	Horas
101	11	40
101	11	55
108	40	67
103	30	90
105	20	120
108	13	8
102	12	30
104	30	24
106	40	18
107	11	8

Dependente					
<u>Cód Empre- gado</u>	<u>Cód Depen- dente</u>	Nome	Sexo	DataNasc	Relação
101	1	João	M	28/02/2000	Filho
101	2	Pedro	M	10/10/2007	Filho
108	3	Sandra	F	01/07/2005	Sogra

Dependente					
105	4	Marcia	F	28/09/1940	Filha
105	5	Sabrina	F	17/04/2003	Filha
107	6	Guilherme	M	09/09/2008	Filho
103	7	Izadora	F	10/12/2012	Filha

Figura 4.4 – Instâncias do Esquema de banco de dados relacional

Dando sequencia nos conceitos de restrição de integridade referencial RIR, torna-se necessário apresentar o conceito de chave-estrangeira (CE). Uma chave-estrangeira (do inglês foreign key) determina a existência de uma integridade referencial aplicada entre duas relações (R1 e R2) e ou, eventualmente, aplicada em uma mesma relação.

Você já aprendeu que normalmente um banco de dados, independentemente da regra de negócio, possui várias relações, essas por sua vez, utilizam “n” restrições de integridade referencial. O DBA (ou projetista de dados) deve conhecer a fundo o significado correto dos atributos existentes nas diversas relações que constituem o esquema de banco de dados, para que se possa obter sucesso na elaboração dessas restrições. Tais restrições, frequentemente, são provenientes dos vários relacionamentos existentes entre as entidades ora representadas pelo esquema de banco de dados. Como exemplo, considere o banco de dados representado pela Figura 4.4. Verifique que na relação EMPREGADO, o atributo CódDept refere-se ao departamento em que cada empregado está associado, ou seja, desempenha suas atividades, dessa maneira, esse atributo é considerado uma chave-estrangeira de EMPREGADO, que tem como propósito, referenciar valores vinculados ao atributo CódDept, esse existente na relação DEPARTAMENTO. Para finalizar esse exemplo, considere que o valor associado ao atributo CódDept independentemente da tupla da relação EMPREGADO, deverá, impreterivelmente, estar associado um valor correspondente à chave-primária da relação DEPARTAMENTO. Esse mesmo valor, eventualmente, pode ser nulo somente se o empregado não estiver vinculado a nenhum departamento. Ainda citando como exemplo a Figura 4.4, note que a

tupla referente ao empregado “Venilton José” se encontra vinculado ao atributo CódDept da relação DEPARTAMENTO, evidenciando que o “Venilton José” está associado ao departamento, ora intitulado de “Recursos Humanos”.

É possível ainda abstrair que uma chave-estrangeira pode referenciar sua mesma relação, descrevendo o que denominamos de uma chave-estrangeira recursiva. A fim de exemplificar graficamente o emprego das restrições de integridade referencial, visualize a Figura 4.5, a qual nos apresenta um esquema relacional com inúmeras restrições de integridade referencial, ora representadas pelo uso de setas direcionais.

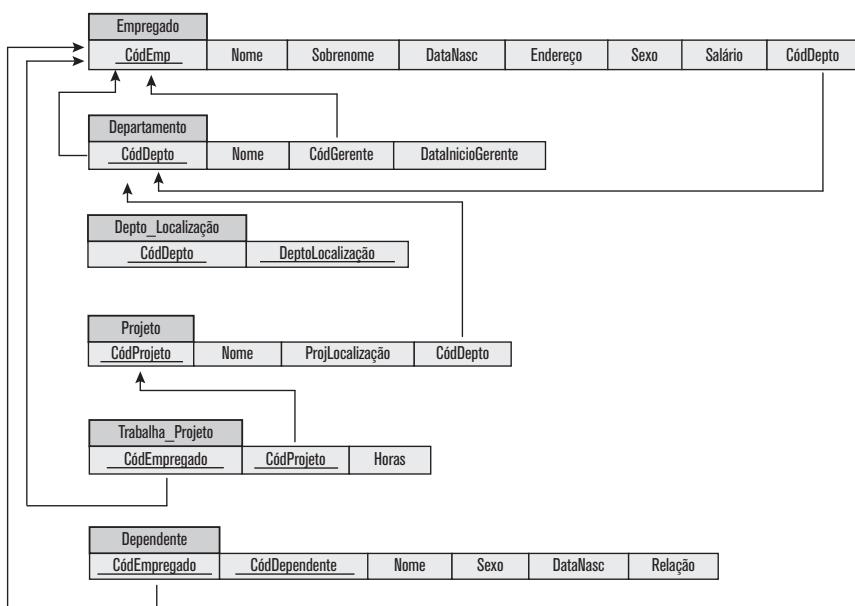


Figura 4.5 – Representação gráfica das restrições de integridade referenciais

Para que seja possível estabelecer a validade das restrições para o banco de dados, essas restrições de integridade devem ser implementadas em um esquema de banco de dados relacional. Sendo assim, torna-se imprescindível o emprego de um Sistema Gerenciador de Banco de Dados Relacional (SGBD-R), sobretudo o uso da linguagem de definição de dados (DDL – Data Definition Language), que por sua vez, promove recursos para a definição dos variados tipos de restrições, assegurando que essa verificação seja realizada pelo SGBD-R de forma automática.

Estudamos até aqui apenas as restrições de integridade de entidade e referencial, porém, outras restrições, a citar, restrições de integridade semântica, também podem ser aplicadas em um banco de dados relacional. Para exemplificar o uso de restrição de integridade semântica, suponha que o salário de um empregado não pode ser superior ao salário do seu diretor e, ou, o número máximo de horas diárias trabalhadas não pode exceder 8 horas.

Para finalizar esse tópico, é importante mencionar a existência de três operações triviais (inserção, alteração e remoção) empregadas em um esquema de banco de dados relacional qualquer. A primeira operação discorre sobre a inserção, a qual é responsável por inserir novas tuplas em uma relação. A segunda operação é a de alteração, onde os valores de um ou mais atributos são alterados, e, por fim, a terceira operação, essa nomeada de remoção, como o próprio nome sugere, realiza a remoção das tuplas. Dessa forma, independente do tipo de operação aplicada em um esquema relacional, as restrições devem ser atendidas a fim de evitar eventuais inconsistências indesejáveis.

4.4 Mapeamento do MER para o Modelo Relacional

Podemos utilizar “n” esquemas relacionais para um esquema ER, isso é, existe diversas formas de se implementar uma modelagem conceitual abstrata.



O MER foi criado para dar subsídio ao processo de modelagem de dados que possui como objetivo realizar a construção de bancos de dados (ROB, 2005).

Dessa forma, nessa etapa, o projetista de dados deve evitar o uso de um grande número de tabelas, sobretudo, evitar que o tempo de resposta seja considerado insatisfatório nas operações de consultas e atualizações de dados. Esse quesito implica em minimizar o número de junções entre as tabelas, evitar atributos opcionais, evitar tabelas subutilizadas, evitar excesso de controles de integridade no banco de dados e evitar organizações de dados em tabelas que possuem um número significante de controle de integridade.

Normalmente, a fase que constitui o processo de mapeamento é formado pelas seguintes etapas:

1. Mapeamento preliminar de entidades e seus atributos
2. Mapeamento de especializações

3. Mapeamento de relacionamentos e seus atributos

Para exemplificar, considere a Figura 4.6, ora representada por uma entidade chamada de EMPREGADOS, que por sua vez, incorpora os atributos RG (atributo identificador) nome e idade. Repare que o processo de mapeamento para esse exemplo é bem simples e objetivo e que, o atributo identificador (chave-primária) é representado pelo uso de um sublinhado no nome do atributo.

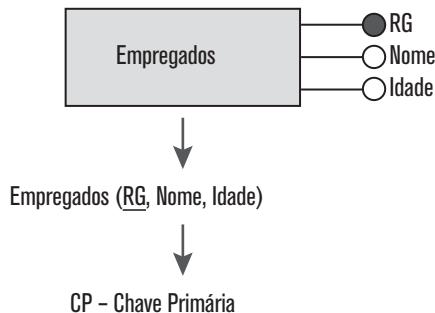


Figura 4.6 – Mapeando convencional de uma entidade

Em relação ao mapeamento de entidades-fracas, deve-se observar que o identificador da entidade considerada forte, constitui parte da chave-primária da tabela correspondente à entidade-fraca e, por outro lado, torna-se chave-estrangeira na tabela fraca. Esse exemplo pode ser visualizado na Figura 4.7. Lembrase que a entidade-fraca nesse exemplo é ITENS. Note que foi utilizada uma chave-primária composta para esse exemplo.

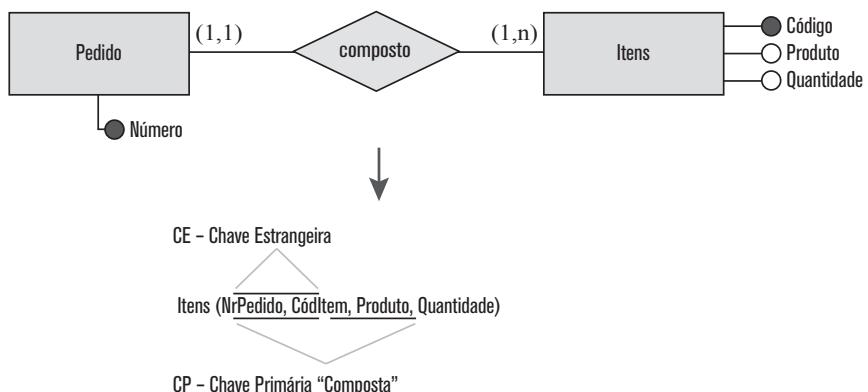


Figura 4.7 – Mapeamento da entidade-fraca (ITENS)

Já o exemplo apresentado pela Figura 4.8, é realizado o mapeamento dos atributos (obrigatório, opcional e composto):

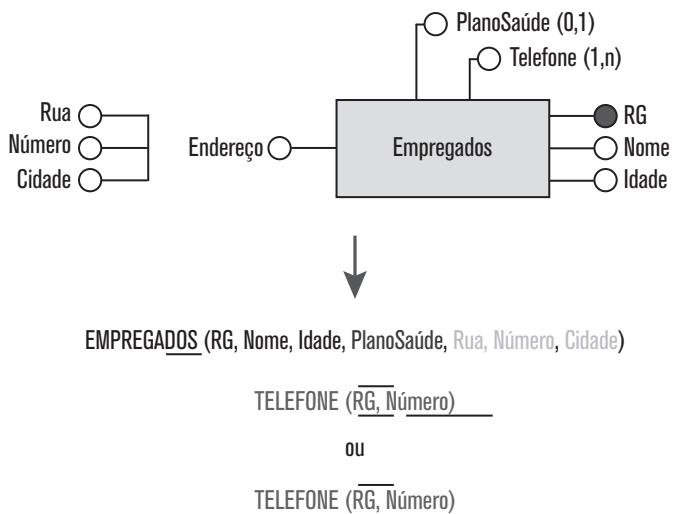


Figura 4.8 – Mapeamento dos atributos (obrigatório, opcional e composto)

Existe também uma outra forma de realizar o mapeamento dos atributos apresentados no exemplo anterior (Figura 4.8). Note que nesse exemplo, ora visualizado pela Figura 4.9, a cardinalidade máxima do atributo telefone é três (FoneRes, FoneCom e Celular).

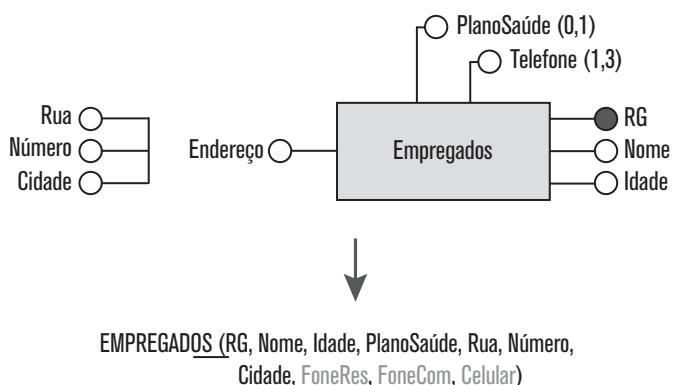


Figura 4.9 – Outra maneira de realizar o mapeamento dos atributos

Agora, a respeito do mapeamento de entidades especializadas, deve-se considerar três opções:

1. Opção: considerar uma única tabela para a entidade considerada genérica e suas especializações;
2. Opção: considerar o uso de tabelas para entidade genérica e para as entidades especializadas;
3. Opção: considerar o uso de tabelas apenas para as entidades especializadas.

No próximo exemplo, a 1^a opção é adotada. Repare na Figura 4.10 que foi realizado o mapeamento da entidade genérica e suas respectivas entidades especializadas, mesclando tudo em uma única tabela. Dessa maneira, o atributo “tipo” pode armazenar mais de um valor apenas se a especialização for não-exclusiva.

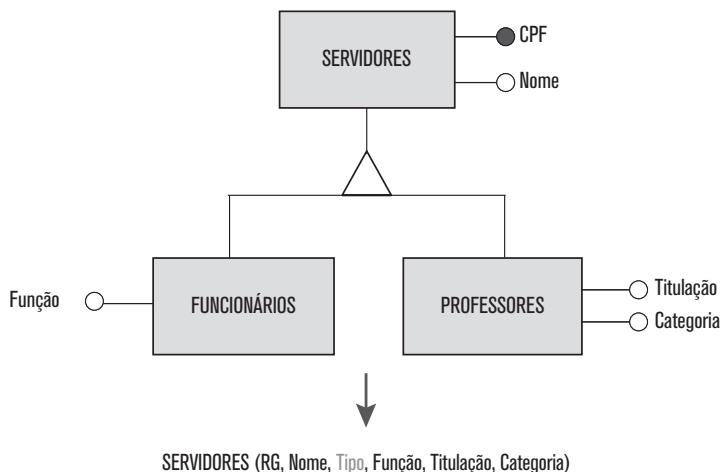


Figura 4.10 – 1^a alternativa: Mapeando da entidade genérica e especializada

Para a 2^a alternativa de mapeamento de entidade genérica e especializada, considere como exemplo a Figura 4.11, a qual foi realizada o mapeamento de forma separada entre a entidade genérica SERVIDORES e suas entidades especializadas, intituladas de FUNCIONARIOS e PROFESSORES.

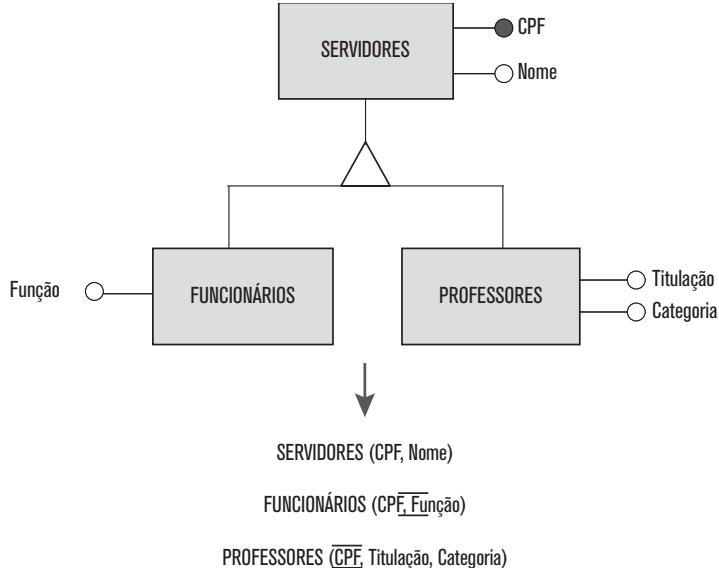


Figura 4.11 – 2^a alternativa: Mapeando de entidade genérica e especializada

Em nossa última alternativa, repare (Figura 4.12) que realizamos apenas o mapeamento das entidades especializadas FUNCIONARIOS e PROFESSORES. Um detalhe importante diz respeito a não utilização dessa alternativa em especializações parciais.

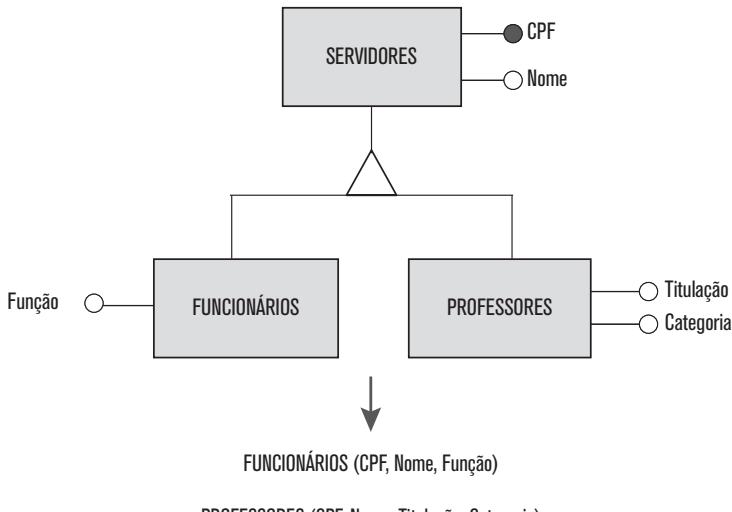


Figura 4.12 – 3^a alternativa: Mapeando apenas as entidades especializadas

O mapeamento dos relacionamentos considera uma análise aplicada na cardinalidade existente nos relacionamentos. Após essa análise, diversas alternativas de mapeamento podem ser escolhidas, a citar: as entidades ora relacionadas podem ser fundidas em uma única tabela; outras tabelas podem ser criadas para acolher os relacionamentos, e, como última alternativa, eventuais chaves-estrangeiras podem ser constituídas nas tabelas para estabelecer corretamente o relacionamento. A Figura 4.13 apresenta um exemplo de mapeamento de um relacionamento, cujo cardinalidade é 1:1 (um-para-um), ou seja, obrigatório em ambos os sentidos.

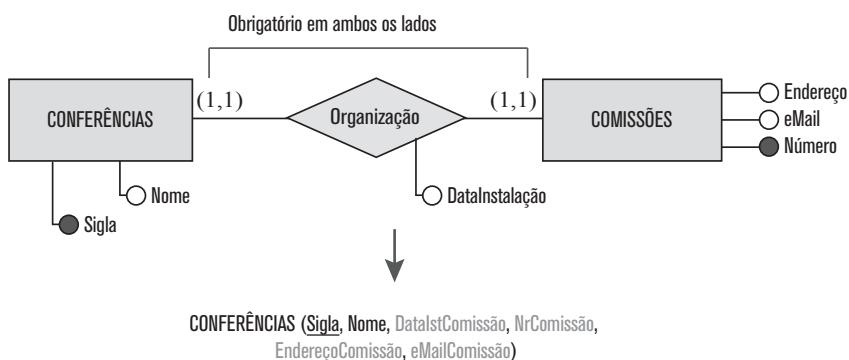


Figura 4.13 – Mapeando um relacionamento (1:1).

Como outro exemplo, podemos tomar o mapeamento de um relacionamento que possui duas cardinalidades, a primeira é (1:1) e a segunda cardinalidade é (0:1), isso simboliza, opcional em um dos sentidos. A Figura 4.14 demonstra o resultado desse mapeamento, perceba que foi similar ao exemplo anterior, ou seja, fundimos em uma única tabela.

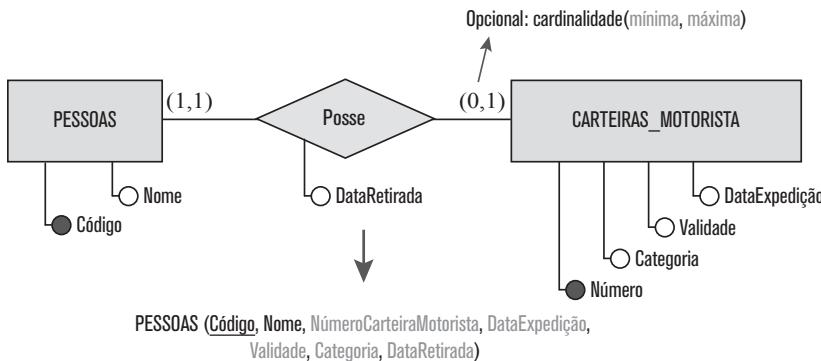


Figura 4.14 – 1^a alternativa: Opcional em um dos sentidos

Existe ainda uma outra alternativa para esse caso, apresentado pela Figura 4.15 que promove também o mapeamento do mesmo MER (opcional em um dos sentidos), todavia, utiliza duas tabelas (**PESSOAS** e **CARTEIRAS_MOTORISTA**).

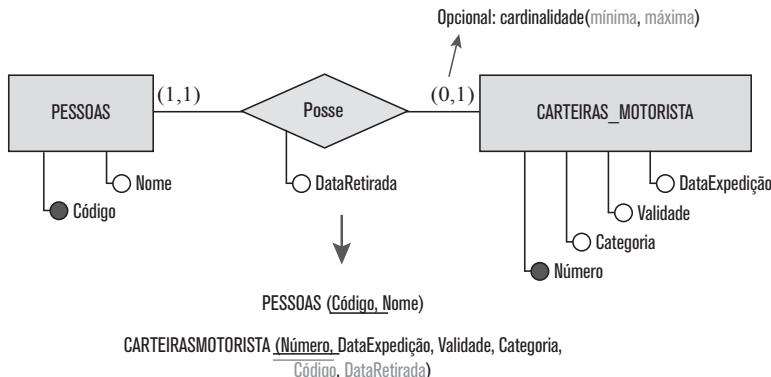


Figura 4.15 – 2^a alternativa: Opcional em um dos sentidos

A Figura 4.16 exemplifica uma alternativa aplicada ao mapeamento de relacionamento opcional em ambos os sentidos, isso é, a cardinalidade mínima das entidades denominadas **HOMENS** e **MULHERES** respectivamente, equivalem à zero, simbolizando um relacionamento opcional independentemente do sentido utilizado.

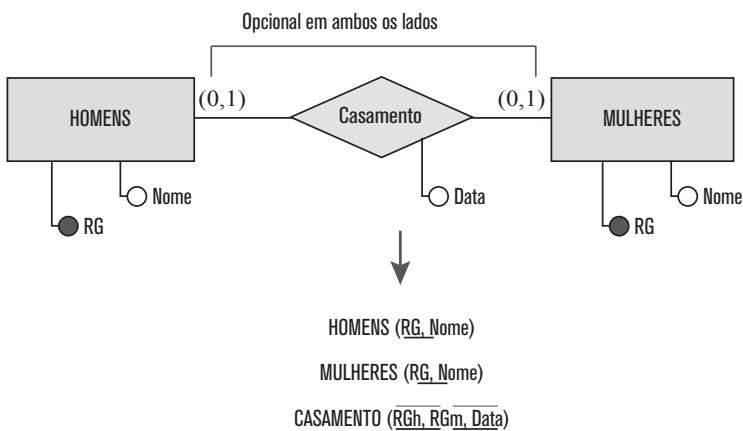


Figura 4.16 – Mapeamento do relacionamento (opcional em ambos os sentidos)

A segunda alternativa para o mesmo MER, representado pela Figura 4.17 considera o mesmo relacionamento anterior, ou seja, opcional em ambos os sentidos, porém realiza o mapeamento apenas das entidades envolvidas.

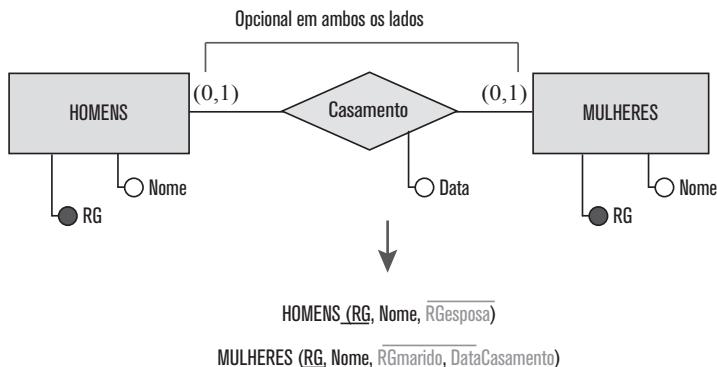


Figura 4.17 – Mapeamento do relacionamento (opcional em ambos os sentidos)

Quando se trata de um relacionamento 1:N, isso é, um relacionamento obrigatório e ou opcional do lado N. O exemplo a seguir, ora representado pela Figura 4.18, é possível identificar que a entidade nomeada de EMPREGADO trabalha com duas alternativas de cardinalidade, uma representando a cardinalidade 1:N e a outra 0:N, a qual caracteriza opcional no lado N. No exemplo apresentado, torna-se possível visualizar o mapeamento final utilizando duas tabelas, essas nomeadas de DEPARTAMENTOS e EMPREGADOS e, ainda, que o atributo referente ao relacionamento LOTAÇÃO se encontra incluído na tabela EMPREGADOS.

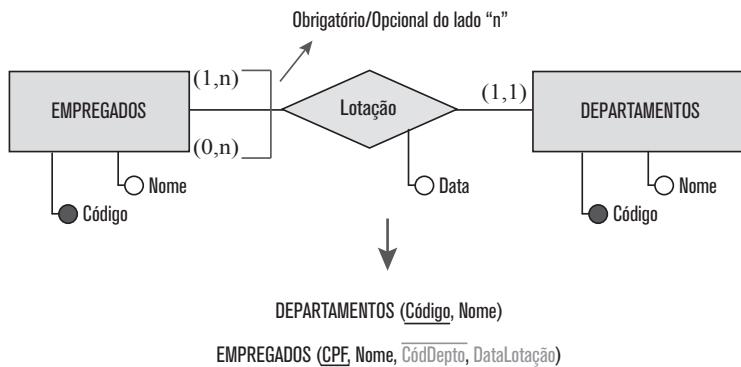


Figura 4.18 – Mapeando do relacionamento (obrigatório e opcional)

Repare agora que, quando trabalhamos com relacionamentos 1:N e opcional do lado “1”, basicamente podemos aplicar duas possibilidades para o processo de mapeamento. Como primeira alternativa, ora representada pela Figura 4.19, é criada três tabelas, duas tabelas destinadas ao mapeamento das entidades **AUTOMÓVEIS** e **PESSOAS**, e uma tabela exclusiva para mapear o relacionamento **POSSE**, que por sua vez, possui como característica os atributos **RG** (chave-estrangeira), **chassi** (chave-primária e chave-estrangeira) e **DataCompra**.

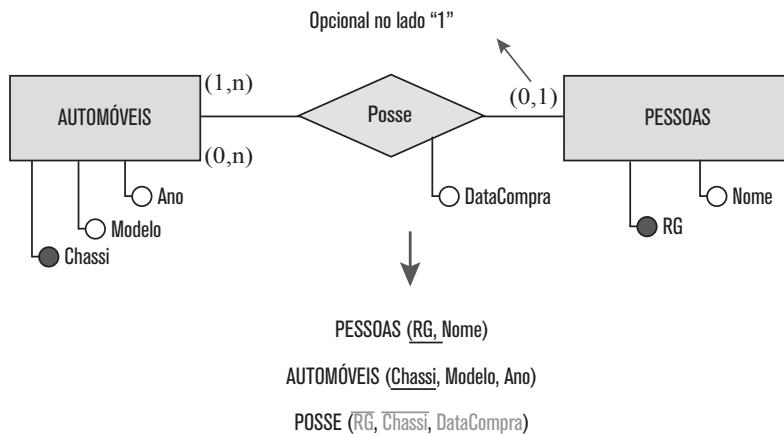


Figura 4.19 – 1^a Opção: Mapeamento do relacionamento (opcional lado “1”).

Como segunda opção para o mapeamento do MER apresentado pela Figura 4.19, é utilizado a Figura 4.20, onde é possível notar que foi utilizado duas tabelas para realizar o mapeamento do relacionamento caracterizado de 1:N, isso é, simplesmente foi criado uma tabela para mapear a entidade PESSOAS e outra tabela para mapear a entidade AUTOMÓVEIS. Repare ainda que inserimos o atributo RG da entidade PESSOAS, que por sua vez é uma chave-estrangeira, com o atributo DataCompra, esse de propriedade do relacionamento POSSE.

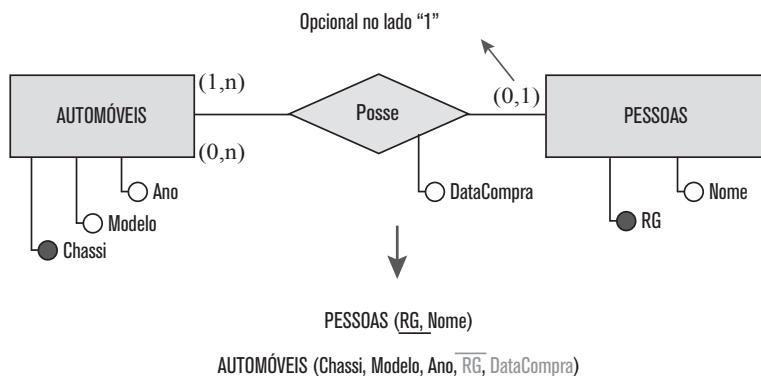


Figura 4.20 – 2^a Opção: Mapeamento do relacionamento (opcional lado “1”).

Para exemplificar um relacionamento N:N, visualize a Figura 4.21. Verifique que temos três tabelas resultantes do processo de mapeamento, ou seja, duas tabelas para mapear as entidades EMPREGADOS e PROJETOS respectivamente, e outra tabela exclusiva para mapear o relacionamento PARTICIPAÇÃO. Dessa maneira, a tabela PARTICIPAÇÃO é formada pelos atributos RG, código e DataInício, e ainda, note que RG e código tornam-se os atributos identificadores constituindo o que denominamos de chave-primária composta. O relacionamento aplicado nesse exemplo, entre as entidades EMPREGADOS e PROJETOS é obrigatório e ou opcional nos dois sentidos.

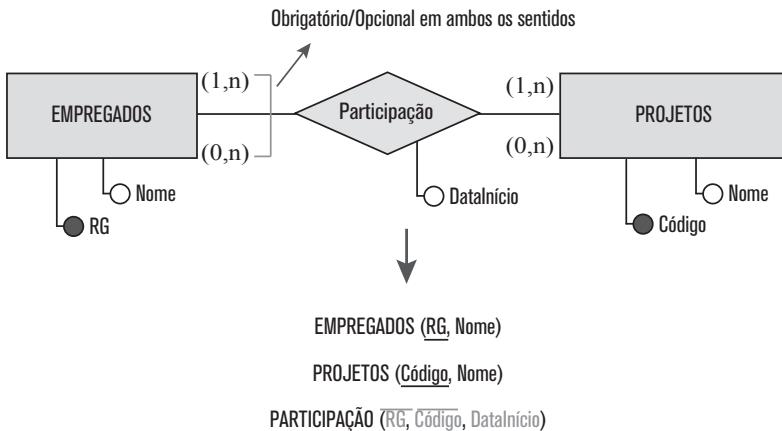
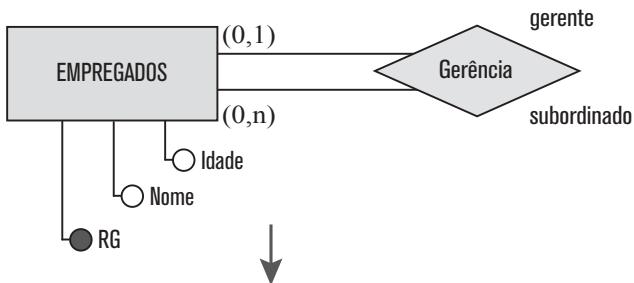


Figura 4.21 – Mapeando um relacionamento N:N (obrigatório e ou opcional nos dois sentidos).

Nesse próximo exemplo, exploraremos conhecimentos referente ao mapeamento de um auto-relacionamento, isso é, de um relacionamento de grau um. Não existe nenhuma exceção, pois é permitido aplicar absolutamente todas as regras de mapeamento vistas até o momento. A Figura 4.22 permite que seja visualizado duas alternativas de mapeamento de um auto-relacionamento. É importante lembrar que para esse tipo de relacionamento é imprescindível o uso de papéis, esses utilizados para identificar adequadamente o papel de cada instância de EMPREGADOS, ora assumindo o papel de GERENTE e ou SUBORDINADO.



1ª Alternativa:

EMPREGADOS (RG, Nome, Idade)

GERÊNCIA (RGempregado, RGgerente)

2ª Alternativa:

EMPREGADOS (RG, Nome, Idade, RGgerente)

Figura 4.22 – Mapeamento de um auto-relacionamento

Em relação ao processo de mapeamento de uma entidade associativa, pode-se utilizar as mesmas recomendações estudadas até o momento. Apenas uma ressalva, é importante identificar corretamente a entidade associativa junto ao MER. A Figura 4.23 exemplifica o mapeamento de entidades associativas.

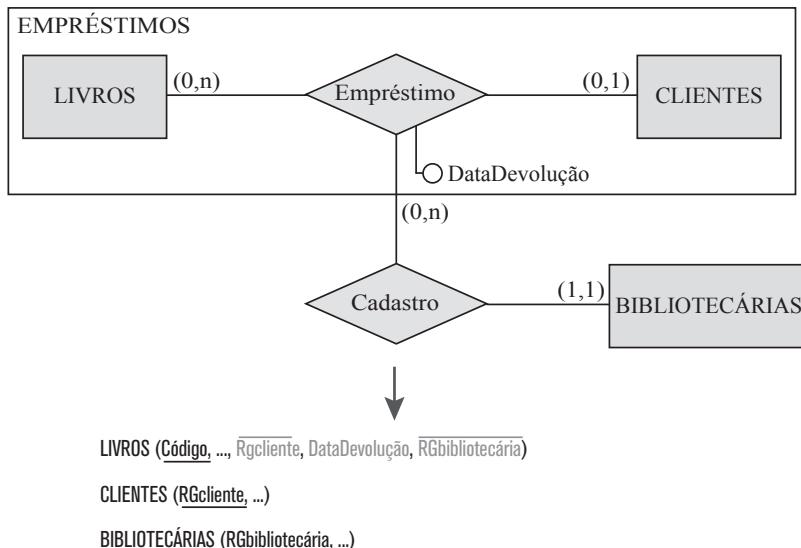


Figura 4.23 – Mapeamento de entidades associativas

Para finalizar esse tópico, estudaremos agora os processos aplicados no mapeamento em relacionamento de grau três. A Figura 4.24 representa um exemplo do processo de mapeamento aplicado em um relacionamento dito como ternário, esse formado pelas entidades INSTITUIÇÕES, PROJETOS e PESQUISADORES. Note que foi considerado a cardinalidade N:N:N que envolve por sua vez as três entidades do MER. Após o mapeamento, repare que foi obtido quatro tabelas, uma destinada para cada entidade e outra exclusiva para o relacionamento intitulado de PESQUISA.

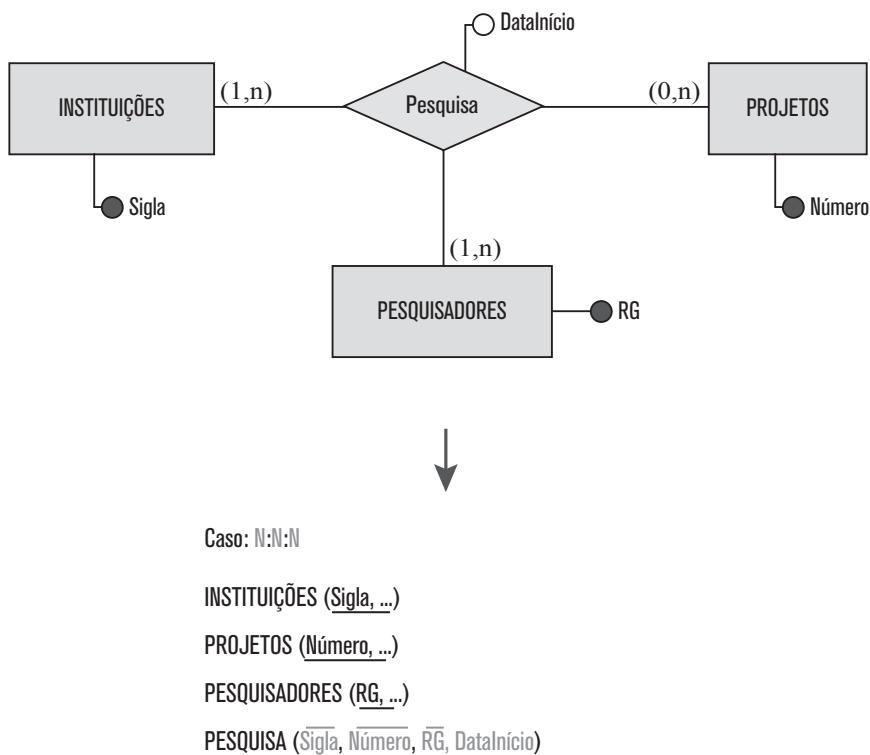
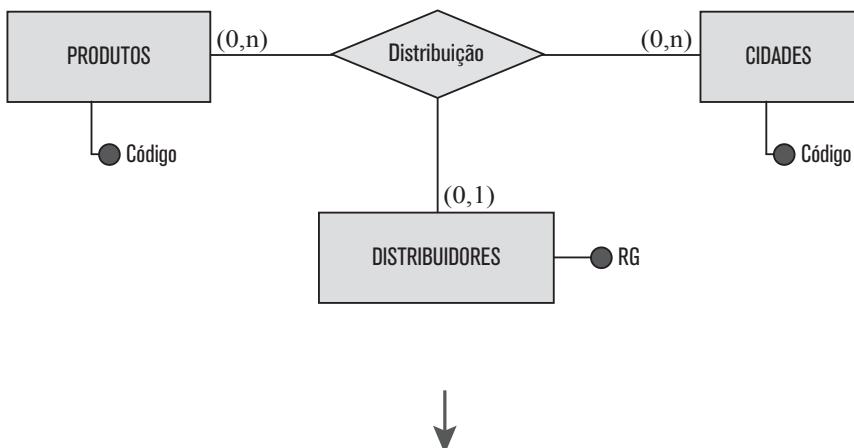


Figura 4.24 – Mapeamento de um relacionamento Grau 3 (N:N:N).

Na Figura 4.25, foi considerado para o exemplo a cardinalidade 1:N:N aplicado em um relacionamento ternário. Como resultado final do processo de mapeamento, foi obtido quatro tabelas, similar ao exemplo apresentado pela Figura 4.24, isso é, uma tabela para mapear cada entidade e outra específica para mapear o relacionamento DISTRIBUIÇÃO. Apenas como distinção do exemplo anterior (Figura 4.24), considere a criação correta da chave-primária composta, que por sua vez, não vinculou o atributo RG chave-estrangeira na chave-primária da tabela DISTRIBUIÇÃO por simplesmente possibilitar que um produto seja distribuído em uma cidade por nenhum ou no máximo um distribuidor.



Caso: 1:N:N

PRODUTOS (Código, ...)

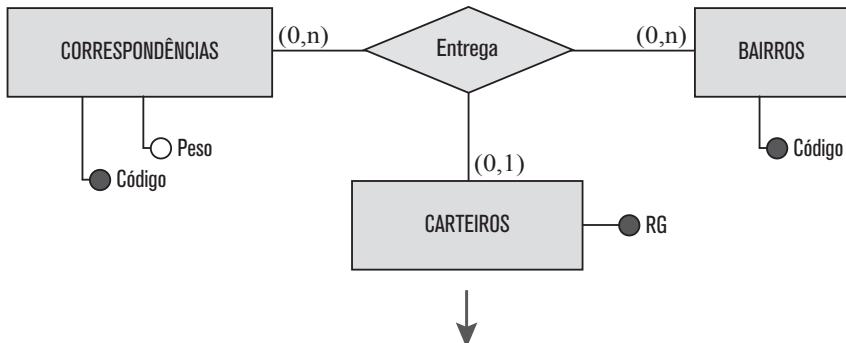
CIDADES (Código, ...)

DISTRIBUIDORES (RG, ...)

DISTRIBUIÇÃO (CódProduto, CódCidade, RG)

Figura 4.25- Mapeamento de relacionamento ternário (1:N:N).

Quando existe relacionamento ternário 1:1:N, esse representado pela Figura 4.26, temos como resultado do processo de mapeamento três tabelas, uma para cada entidade. A fim de atender corretamente o relacionamento presente entre as entidades, note que na tabela CORRESPONDENCIAS existe as chaves-estrangeiras CódBairro e RG, as quais determinam que uma correspondência será entregue única e exclusivamente por um carteiro para um único bairro.



Caso: 1:1:N

BAIRROS(Código, ...)

CARTEIROS (RG, ...)

CORRESPONDÊNCIAS (CódCarta, Peso, CódBairro, RG, ...)

Figura 4.26 – Mapeamento do relacionamento ternário (1:1:N).

Para concluir esse tópico, que tratou acerca do mapeamento MER para o modelo de dados relacional, o último exemplo de mapeamento é apresentado pela Figura 4.27, que apresenta um relacionamento ternário 1:1:1. Repare que quando lidamos com esse tipo de relacionamento, o resultado do processo de mapeamento inclui apenas uma tabela, essa responsável por incorporar todos os atributos das entidades envolvidas no relacionamento. Para esse exemplo, foi gerado uma tabela nomeada de **VEICULO**, que por sua vez, incluiu os atributos Código, PesoPainel, CódMotor, FabrMotor, CódLataria e ModLataria.

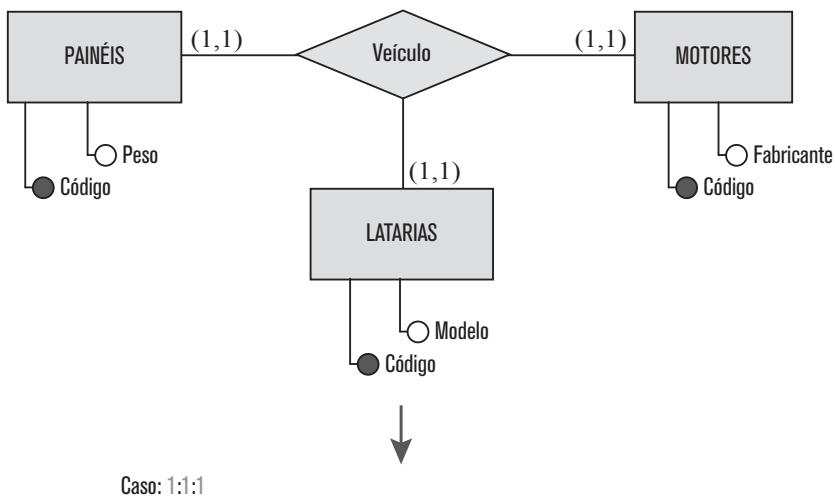


Figura 4.27 – Mapeamento do relacionamento ternário (1:1:1).



ATIVIDADE

1. Conceitue chave-primária simples e composta. Dê pelo menos três exemplos de cada tipo de chave-primária.
2. Utilize suas próprias palavras para discorrer sobre os conceitos de Restrição de Integridade de Entidade (RIE) e Restrição de Integridade Referencial (RIR). Cite exemplos para ambos os conceitos.
3. Analise os requisitos a seguir e constitua sua modelagem relacional que atenda algumas necessidades de informação, a citar: Qual o código e a descrição de cada projeto desempenhado na empresa? Qual é o número da matrícula e nome de cada funcionário? Quais são as possíveis funções desempenhadas na empresa?
Imagine que uma empresa os funcionários trabalham em projetos, onde em cada projeto um funcionário poderá exercer diversas funções de acordo com as regras expostas abaixo:
 - Os funcionários podem realizar distintas funções em diversos projetos;
 - Eventualmente, um funcionário pode exercer em um mesmo projeto distintas funções;

- Em um determinado projeto podemos ter uma mesma função (atribuição) exercida por distintos funcionários;
 - Por outro lado, um funcionário poderá realizar a mesma função em distintos projetos;
4. Quais características são desejáveis para uma chave-candidata?
-

REFLEXÃO

Parabéns! Finalizamos mais um capítulo de estudo!

Neste capítulo, trabalhamos muito, sobretudo por abordamos assuntos mais áridos. Esperamos que tenha compreendido os conteúdos.

Neste capítulo estudamos o modelo de dados relacional e seus componentes, como por exemplo, como as chaves funcionam e qual a sua relevância como atributos identificadores, e ainda, seu principal objetivo na identificação única de instâncias de entidades.

Na sequência, estudamos algumas restrições de integridade, consideradas primordiais para impor a consistência dos dados em qualquer esquema de banco de dados.

A partir de agora, é fortemente indicado à realização dos exercícios sugeridos, e, se possível, realize pesquisas além do conteúdo da apostila para incrementar seus conhecimentos. Caso, eventualmente, você se deparar com alguma dúvida, volte aos tópicos e refaça a leitura com bastante concentração..

LEITURA

Artigos on-line: para que você possa aumentar ainda mais o seu nível de aprendizagem referente ao modelo de dados relacional:

<http://imasters.com.br/artigo/2419/banco-de-dados/o-modelo-relacional-de-dados-parte-01/>
<http://imasters.com.br/artigo/2419/banco-de-dados/o-modelo-relacional-de-dados-parte-02/>
<http://imasters.com.br/artigo/2419/banco-de-dados/o-modelo-relacional-de-dados-parte-03/>

Livros:

Sistemas de Banco de Dados: Projeto, Implementação e Administração. ROB e CORONEL.
Sistemas de Bancos de Dados. Elmasri e Navathe;

Nesses dois livros é possível encontrar com detalhes todo o conteúdo explorado nesta unidade. É fortemente recomendado que você maximize seus conhecimentos nestas bibliografias.



REFERÊNCIAS BIBLIOGRÁFICAS

ELMASRI, R.; NAVATHE, S.B. Sistemas de bancos de dados. São Paulo: Pearson (Addison Wesley), 2005.

KORTH, H.; SILBERCHATZ, A. Sistemas de bancos de dados. 3. ed. São Paulo: Makron Books, 1998.

HEUSER, C. A. Projeto de Bancos de Dados. 2. ed. Porto Alegre: Sagra Luzzatto, 1999.

PRESSMAN, R.S. Engenharia de software. São Paulo: Makron Books, 1995.

ROB, P; CORONEL, C. Sistemas de Banco de Dados: Projeto, Implementação e Administração. 8. Ed. São Paulo: Cengage Learning, 2011.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: Elsevier, 2006.

DATE, C. J.; Introdução a Sistemas de Banco de Dados; 8^a. Ed.; Trad. Daniel Vieira; Rio de Janeiro: Elsevier, 2003.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4 ed. Instituto de Informática da UFRGS, Sagra DC Luzzatto, 1998.

RAMAKRISHNAN, R. GEHRKE, J. Database Management Systems. 2. ed. Boston: McGraw-Hill, 2000.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T.; JAGADISH, H. V.; Database Modeling and Design – Logical Design. 5^a ed., Burlington – USA: Elsevier, 2011.

CASTRO, E. B.; Modelagem Lógica de Dados: construção básica e simplificada; Rio de Janeiro: Editora Ciência Moderna, 2012.

PUGA, S.; FRANÇA, E.; GOYA, M.; Banco de Dados – Implementação em SQL, PL/SQL e Oracle 11g; São Paulo: Pearson Education do Brasil, 2013.



NO PRÓXIMO CAPÍTULO

No próximo capítulo, estudaremos sobre o processo de normalização, conceito também considerado importantíssimo para obtermos projetos de banco de dados íntegros e concisos!

5

Normalização

5 Normalização

No capítulo anterior, aprendemos sobre o modelo de dados relacional, regras de integridade de entidade e referencial. Vamos agora aumentar nossos conhecimentos num assunto de extrema relevância para a criação de projetos de banco de dados bem estruturados.

Uma vez abstraídos os principais conceitos relacionados ao modelo de dados relacional, você não deve ter se surpreendido pelo fato desse modelo ser amplamente adotado, sendo considerado um clássico na maioria das literaturas que discorrem sobre o assunto.

Neste capítulo, estudaremos sobre a normalização, o que é, e para que aplicá-la. Explicar o que é uma tabela aninhada e aprender sobre as principais dependências funcionais.

Para fechar a unidade, estudaremos com exemplos práticos as cinco formas normais (1FN, 2FN, 3FN, 4FN e 5FN), enfatizando as três primeiras, essas consideradas como inevitáveis.

Para que seja possível estudarmos de forma real todas as fases que envolvem o processo de normalização, devemos abstrair corretamente quando e como as dependências funcionais ocorrem. Para isso, a partir de agora, fique atento quando abordamos conceitos sobre dependências funcionais.



OBJETIVOS

O propósito dessa unidade é estudarmos o processo de normalização de dados, de modo a compreendermos suas principais fases. A partir desse conceito, deveremos estar aptos para representar tabelas livres de redundâncias e ou qualquer tipo de inconsistências.



REFLEXÃO

Você se lembra de nossa discussão no capítulo 4 sobre os conceitos do modelo de dados relacional?

Estudamos nos capítulos anteriores sobre os níveis de abstração, você ainda se lembra?

Em relação às chaves-primárias, estrangeiras e regras de integridade, você é capaz de defini-los?

5.1 Normalização

Em algumas ocasiões, aplicações computacionais podem apresentar problemas de desempenho oriundos de uma modelagem de dados mal realizada, a qual acaba produzindo uma quantidade expressiva de redundância de dados, esses por sua vez prejudicam o sucesso do objetivo empresariais.

Erros na abstração da estrutura do banco de dados constituem barreiras no que se refere a resolução desse tipo de problema. O processo de normalização tem como objetivo promover a reestruturação dos dados a fim de eliminar qualquer tipo de redundância caracterizada como indesejável, ora presentes em alguns esquemas de banco de dados.

Basicamente, o processo de normalização é constituído por cinco formas normais, entretanto, se for considerado as três primeiras formas normais, já é possível alcançarmos um modelo de banco de dados considerado bem estruturado e conciso.

Em meados dos anos 70, quando Codd criou os primeiros conceitos que discorriam sobre o processo de normalização, pouco se conhecia sobre seu propósito. Inúmeros profissionais na ocasião desconheciam a maneira de se aplicar normalização de dados. Varias dúvidas pertinentes ao tema surgiram, a citar: O que efetivamente se almeja no processo de normalização de dados?

A normalização de dados pode ser considerada como um processo onde o objetivo é examinar minuciosamente as estruturas e tabelas de um banco de dados relacional, minimizando possíveis redundâncias e inconsistências de dados, assim promovendo a eliminação das irregularidades de inserção, alteração e exclusão.

As formas normais que constituem o processo de normalização de dados são utilizadas em um banco de dados relacional como um filtro aplicado sobre as entidades e seus respectivos relacionamentos, eliminando elementos indesejáveis sem ocasionar perda de informação.

Ao se conduzir um processo de normalização de dados em um banco de dados relacional, desejamos ao final desse processo evitar: eventuais duplicações de dados (atributos multivvalorados); qualquer tipo de dependência funcional e redundância de dados que possam conduzir a perdas de informação.

As fases do processo de normalização de dados devem ser realizadas de maneira contínua (em cascata), ou seja, inicialmente deve-se aplicar a primeira forma normal (1FN), na sequência, deve-se aplicar a segunda forma-normal

(2FN), logo após, deve-se conduzir a terceira forma-normal (3FN), posteriormente, a quarta forma-normal (4FN) e, para finalizar o processo, deve-se aplicar a quinta forma-normal (5FN). Conceitualmente, a forma normal mais elevada possui melhor eficiência se comparamos com sua antecessora, isto é, a 5FN é mais eficaz que a 4FN, que por sua vez, é melhor que a 3FN, e assim, sucessivamente. É importante mencionar que, apesar do processo de normalização de dados ser constituído por cinco formas normais, normalmente, em cenários empresariais convencionais, realizamos a aplicação das três primeiras regras de normalização de dados, ou seja, 1FN, 2FN e 3FN, resultando em um esquema de banco de dados considerado aceitável. As regras que constituem a 4FN e 5FN tornam-se aconselhável apenas em estrutura de banco de dados que realmente carecem da aplicação de tais regras.

5.1.1 Aninhamento de Tabela

Uma tabela não normalizada (aninhada) é uma tabela que não atende a nenhuma das regras da 1FN, e, na maioria das vezes, essa tabela pode possuir uma ou mais tabelas ditas como aninhadas (uma tabela dentro de outra tabela, e assim, sucessivamente).



ATENÇÃO

Regras de Normalização: as regras de normalização podem ser consideradas como sendo uma formalização dos bons princípios para o projeto de bancos de dados ou ainda como regras de conduta na definição de registros em um projeto de banco de dados. (CASTRO, 2012).

Uma tabela é dita como aninhada quando possui grupo de dados redundantes, colunas multivvaloradas, colunas não atômicas, isso é, não possui dados atômicos, e sim, tabelas inclusas dentro de outras. Para apresentar um exemplo de tabela aninhada (não normalizada) considere a Figura 5.1.

CódProjeto	Tipo	Descrição	Cód Empregado	Nome	Categoria	Salário	Empregado	
							Data início	Tempo
PSI001	Desenvolvimento de Software	Sistema ERP	6124	Victor Nunes	A1	2	11/01/2001	48
			5134	Fábio Cardoso	A2	2	10/02/2001	48
			4211	José Ribeiro	C1	4	10/03/2002	36
			6162	Carlos da Silva	A2	2	10/04/2002	36
			1189	Sonia Purcini	A1	2	11/01/2002	24
			1189	Sonia Purcini	A1	2	05/01/2003	24
PSI034	Manutenção	Sistema Apoio à Decisão	4211	José Ribeiro	C1	2	01/04/2001	48
			2141	João de Freitas	A2	4	11/01/2002	24

Figura 5.1 – Exemplo de aninhamento de tabelas

Pode-se identificar a existência de aninhamento entre tabelas, sobretudo por ser fácil de identificar duas tabelas aninhadas, ou seja, uma tabela com os dados referente aos empregados e outra tabela ora armazenando os dados acerca dos projetos desempenhados por esses mesmos empregados. Caso seja aplicado o processo de mapeamento da tabela utilizada como exemplo (Figura 5.1), seria obtido como resultado final o esquema de dados abaixo:

PROJETOS (CódProjeto, Tipo, Descrição, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)

Com base nesse esquema ora não normalizado, iniciaremos o processo de normalização de dados, aplicando as regras necessárias para que essa tabela possa ser considerada como bem projetada, livre de redundância de dados e eventuais inconsistências, ou seja, sem a presença do que denominamos anomalias de inserção, alteração e remoção.

O processo de normalização de dados deve seguir obrigatoriamente a hierarquia apresentada pela Figura 5.2.

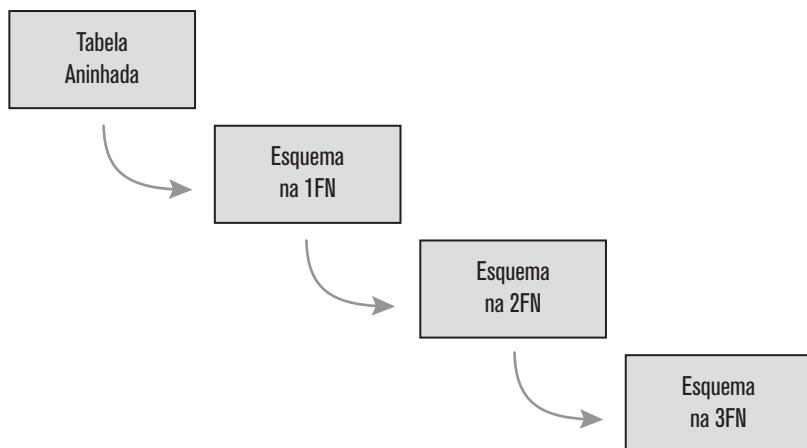


Figura 5.2 – Principais regras de normalização de dados

5.1.2 Regra: Primeira forma normal (1FN)

As regras referente a primeira forma normal fornecesse subsídios para iniciar o processo de normalização de dados. Tendo como referência a tabela aninhada, a partir dessa, será aplicada a 1FN. Uma tabela se encontra na 1FN quando todas as suas tuplas são consideradas distintas entre si, e, absolutamente, não poderá existir nenhum tipo de aninhamento de tabelas. Conclúisse que, uma tabela está na 1FN quando todos os valores atrelados as suas colunas são consideradas monovalorados.

A transformação de uma tabela não normalizada, ou dita também como aninhada em um esquema que atenda as regras pertinentes a 1FN envolve basicamente duas alternativas: lançar uso de uma única tabela aninhada contendo redundância de dados ou, constituir uma tabela para cada tabela aninhada. Sem dúvida, a segunda alternativa é mais conveniente, entretanto, apresentaremos as duas possibilidades para que você analise suas vantagens e desvantagens.

Como primeira alternativa do processo de normalização que envolve as regras da 1FN, tem-se o uso de apenas uma única tabela onde os registros das linhas externas à tabela aninhada são repetidos para cada linha da tabela aninhada.

– **Tabela Não-Normalizada (Aninhada):**

PROJETOS (CódProjeto, Tipo, Descrição, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)



– **1FN (1^a Alternativa):**

PROJETOS (CódProjeto, Tipo, Descrição, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)

Perceba que os registros pertinentes ao projeto são duplicados para cada empregado que trabalha no projeto.

Já a segunda alternativa para a aplicação da 1FN na tabela aninhada utilizada como exemplo, cria-se uma tabela para a própria tabela que está sendo normalizada e uma tabela para representar cada tabela considerada aninhada.

– **Tabela Não-Normalizada (Aninhada):**

PROJETOS (CódProjeto, Tipo, Descrição, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo).



– **1FN (2^a Alternativa)**

PROJETOS (CódProjeto, Tipo, Descrição).

PROJETOS_EMPREGADOS (CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)

Note que a primeira alternativa pode ser considerada como válida, mesmo que o resultado final contenha apenas uma única tabela. A segunda alternativa, que por sua vez, fragmenta uma tabela aninhada em várias outras tabelas, em um primeiro momento, pode ser considerada arriscada por refletir eventuais perdas de dados. Todavia, em cenários empresariais reais, existe uma preferência de utilizar as regras que compõe a segunda alternativa, isso é, empregar o fracionamento de tabelas.

Em nosso cotidiano, podemos nos confrontar com a existência de diversas tabelas aninhadas, essas possuindo vários níveis de aninhamento, impedindo à visualização adequada da tabela na 1FN, se considerarmos a primeira alternativa, a qual emprega exclusivamente apenas uma única tabela.

Com o propósito de detalharmos o processo pertinente a criação de uma tabela que conteemple as regras da 1FN, a partir de uma tabela aninhada, dividimos em fases como descritas a seguir:

- 1^a Fase: a chave-primária da tabela na 1FN é basicamente igual à chave-primária da tabela aninhada (não normalizada).

- **Tabela Não-Normalizada (Aninhada):**

(CódProjeto, Tipo, Descrição, (CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)).



- **1FN:**

- 2^a Fase: constituir uma nova tabela, essa constituída pelas chaves-primárias de cada uma das tabelas na qual a tabela em questão se encontra aninhada, e, outra tabela constituída pelas colunas da própria tabela aninhada.

- **Tabela Não-Normalizada (Aninhada):**

(CódProjeto, Tipo, Descrição, (CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)).



- **1FN:**

(CódProjeto, Tipo, Descrição) (CódProjeto, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo).

- 3^a Fase: estabelecer, na 1FN, as chaves-primárias das tabelas correspondentes à tabela aninhada. Nessa fase então, para a tabela externa, simplesmente transcrevemos a chave-primária para a tabela na 1FN.

– **Tabela Não-Normalizada (Aninhada):**

(CódProjeto, Tipo, Descrição, (CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)).

– **1FN:**

(CódProjeto, Tipo, Descrição) (CódProjeto, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)).

Agora é possível analisar o resultado final produzido pela 3^a fase. Você deve estar se questionando: Em relação a chave-primária, qual seria a opção considerada ideal para tabela PROJETOS_EMPREGADOS? Para escolhermos uma chave-primária adequadamente, torna-se necessário identificar com qual frequência os valores vinculados ao atributo CódEmpregado aparece na tabela considerada aninhada. Apenas uma vez ou várias vezes? Retorne na Figura 5.1 e obtenha essa informação.

Não é necessário muito trabalho para certificar que um empregado, eventualmente, pode trabalhar em diversos projetos. Para exemplificar, os empregados Sonia Purcini e José Ribeiro, estão alocados em dois projetos, esses descritos como Sistema ERP e Sistema de Apoio à Decisão. Dessa forma, os valores associados à coluna CódEmpregado (chave-primária da tabela aninhada), de forma inoportuna, se apresenta diversas vezes na tabela aninhada.

Sendo assim, torna-se necessário a associação entre as colunas CódEmpregado e CódProjeto para identificar exclusivamente as diversas ocorrências, constituindo dessa maneira, o que denominamos de chave-primária composta.

– **1FN:**

(CódProjeto, Tipo, Descrição) (CódProjeto, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo))

Após utilizarmos as regras para contemplar a 1FN em nossa tabela aninhada, obteríamos como resultado duas tabelas, conforme apresentada na Figura 5.3.

<u>CódProjeto</u>	<u>Tipo</u>	<u>Descrição</u>
PSI001	Desenvolvimento de Software	Sistema ERP
PSI034	Manutenção	Sistema Apoio à Decisão

Empregado						
<u>CódProjeto</u>	<u>CódEmpregado</u>	Nome	Cate-goria	Salário	Data início	Tempo
PIS001	6124	Victor Nunes	A1	2	11/01/2001	48
PIS001	5134	Fábio Cardoso	A2	2	10/02/2001	48
PIS001	4211	José Ribeiro	C1	4	10/03/2002	36
PIS001	6162	Carlos da Silva	A2	2	10/04/2002	36
PIS001	1189	Sonia Purcini	A1	2	11/01/2002	24
PSI034	1189	Sonia Purcini	A1	2	05/01/2003	24
PSI034	4211	José Ribeiro	C1	2	01/04/2001	48
PSI034z	2141	João de Freitas	A2	4	11/01/2002	24

Figura 5.3 – Tabelas resultantes da 1FN (a partir de uma tabela aninhada).

5.1.3 Regra: Segunda Forma Normal (2FN)

Para que seja possível estudarmos as regras pertinentes a 2FN e 3FN, torna-se crucial o entendimento adequado do conceito de dependência funcional. Considere uma tabela relacional qualquer, dizemos que uma coluna C_2 depende funcionalmente de uma coluna C_1 , isso é, uma coluna C_1 por sua vez, determinada C_2 , ou ainda, para todas as tuplas da tabela, para cada valor de C_1 presente na tabela, aparece o mesmo valor de C_2 . A Figura 5.4 demonstra um exemplo de dependência funcional existente entre as colunas Código e Salário. A interpretação para essa dependência funcional seria: salário depende funcionalmente de código.

...	Código	...	Salário	...
...	Emp01	...	R\$6.059,00	...
...	Emp03	...	R\$6.059,00	...
...	Emp01	...	R\$6.059,00	...
...	Emp02	...	R\$3.750,00	...
...	Emp02	...	R\$6.059,00	...
...	Emp03	...	R\$3.750,00	...
...	Emp01	...	R\$6.059,00	...

Código → Salário

Figura 5.4 – Dependência funcional entre as colunas Salário e Código

A próxima Figura 5.5, é factível identificar a ausência da dependência funcional entre as colunas intituladas de A e B.

A	B	C	D
B	5	2	20
C	4	2	15

B	6	7	20
B	5	2	20
C	2	2	15
C	4	2	15
A	10	5	18
A	12	3	18
A	10	5	18
B	5	2	20
C	4	2	15
A	10	5	18
C	4	2	15

Figura 5.5 – Ausência de dependência funcional (A e B)

A seguir, será apresentado outro exemplo de dependência funcional, ora apresentado na Figura 5.6, onde a coluna “C” depende funcionalmente de uma combinação de colunas (A,B).

A	B	C	D
B	5	2	20
C	4	2	15
B	6	7	20
B	5	2	20
C	2	2	15
C	4	2	15
A	10	5	18

A	12	3	18
A	10	5	18
B	5	2	20
C	4	2	15
A	10	5	18
C	4	2	15

$$(A, B) \rightarrow C$$

Figura 5.6 – Dependência funcional (“C” depende funcionalmente da associação das colunas “A” e “B”).

Agora que você já aprendeu o conceito de dependência funcional e, a partir de agora, é capaz de identificá-lo em uma tabela relacional, podemos avançar em nossos estudos aprendendo as novas regras da 2FN. A 2FN tem como propósito eliminar um determinado tipo de redundância de dados. Para exemplificar, considere o esquema a seguir:

(CódProjeto, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo).

É possível identificarmos que os dados referente ao empregado (Nome, Categoria e Salário) são considerados redundantes para os empregados que trabalham em mais de um projeto. A Figura 5.7 nos apresenta um exemplo dessa redundância.

Empregado						
CódProjeto	Cód Empregado	Nome	Cate-goria	Salário	Data início	Tempo
PIS001	6124	Victor Nunes	A1	2	11/01/2001	48
PIS001	5134	Fábio Cardoso	A2	2	10/02/2001	48
PIS001	4211	José Ribeiro	C1	4	10/03/2002	36

PIS001	6162	Carlos da Silva	A2	2	10/04/2002	36
PIS001	1189	Sonia Purcini	A1	2	11/01/2002	24
PSI034	1189	Sonia Purcini	A1	2	05/01/2003	24
PSI034	4211	José Ribeiro	C1	2	01/04/2001	48
	2141	João de Freitas	A2	4	11/01/2002	24

Figura 5.7 – Redundância aplicada as colunas (nome, categoria e salário).

Uma tabela é dita na 2FN, quando, além de se enquadrar na 1FN, deve estar livre de dependência funcional parcial.

Bem, você deve estar se perguntando. Afinal, o que é uma dependência funcional parcial? Uma dependência funcional parcial é formada quando uma determinada coluna depende apenas de parte de uma chave-primária composta. Por meio do esquema a seguir, é factível a identificação de uma dependência funcional parcial.

– 1FN:



Note que as colunas nome, categoria e salário dependem por sua vez apenas de parte da chave-primária composta, isso é, apenas da coluna CódEmpregado, não tornando necessário a associação entre as duas colunas (CódProjeto e CódEmpregado) para identificar exclusivamente os valores pertinentes as colunas nome, categoria e salário.

Com o propósito de distinguir uma dependência funcional parcial de uma dependência funcional não parcial, considere o esquema utilizado acima, entretanto, verifique que destacamos apenas as colunas que dependem de toda a chave-primária composta.



As colunas DataInício e Tempo dependem funcionalmente de toda a chave-primária composta, constituídas pelas colunas CódProjeto e CódEmpregado.

Um detalhe relevante que devemos destacar é que, se uma tabela qualquer se encontra na 1FN e possui apenas uma coluna como chave-primária (chave-primária simples), a mesma não possui dependências funcionais parciais, pois é impossível uma coluna depender de uma parte de uma chave-primária simples. Sendo assim, considere que toda a tabela que contempla as regras da 1FN e que possui apenas uma coluna como chave-primária já se enquadra na 2FN automaticamente.

Com o objetivo de exemplificar, apresentamos o esquema a seguir, onde é possível constatar que a tabela intitulada de PROJETOS já atende as regras da 2FN, sobretudo por não possuir dependências funcionais parciais (chave-primária simples).

- 1FN:

(CódProjeto, Tipo, Descrição) (CódProjeto, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)

- 2FN:

(CódProjeto, Tipo, Descrição)

A tabela que possui a chave-primária composta e colunas não chave deve ser examinada com cuidado. Verifique cuidadosamente o esquema a seguir, o qual já contempla as regras da 1FN.

- 1FN:

Projetos_Empregados (CódProjeto, CódEmpregado, Nome, Categoria, Salário, DataInício, Tempo)

Nesses casos, deve-se realizar o seguinte questionamento para cada coluna não chave: a coluna depende de toda a chave-primária composta ou apenas de

parte dela? ou, para identificar exclusivamente um valor da coluna é necessário o uso de toda a chave-primária ou apenas de parte dela?

Posteriormente essa verificação, colunas identificadas como dependentes de toda a chave-primária composta devem permanecer na tabela original, conforme apresentado no esquema a seguir:

- **1FN:**



- **2FN:**

Projetos_Empregados (CódProjeto, CódEmpregado, DataInício, Tempo)

Como resultado final, ou seja, posteriormente aplicar as regras referente a 2FN, o esquema de banco de dados pode ser resumido como:

- **2FN:**

Projetos (CódProjeto, Tipo, Descrição)

Projetos_Empregados (CódProjeto, CódEmpregado, DataInício, Tempo)

É possível ainda representar o esquema de banco de dados anterior por meio do uso de tabelas, essas identificadas por sua vez de Projetos, Empregado e Projetos_Empregados, conforme a Figura 5.8.

Projetos		
<u>CódProjeto</u>	Tipo	Descrição
PSI001	Desenvolvimento de Software	Sistema ERP
PSI034	Manutenção	Sistema Apoio à Decisão

Tabela: Projetos_empregados			
<u>CódProjeto</u>	Cód Empregado	Data início	Tempo
PIS001	6124	11/01/2001	48
PIS001	5134	10/02/2001	48
PIS001	4211	10/03/2002	36
PIS001	6162	10/04/2002	36
PIS001	1189	11/01/2002	24
PSI034	1189	05/01/2003	24
PSI034	4211	01/04/2001	48
PSI034	2141	11/01/2002	24

Tabela: Empregados			
<u>Cód Empregado</u>	Nome	Categoria	Salário
6124	Victor Nunes	A1	2
5134	Fábio Cardoso	A2	2
4211	José Ribeiro	C1	4
6162	Carlos da Silva	A2	2

1189	Sonia Purcini	A1	2
1189	Sonia Purcini	A1	2
4211	José Ribeiro	C1	2
2141	João de Freitas	A2	4

Figura 5.8 – Tabelas resultantes do processo de normalização (2FN).

5.1.4 Regra: Terceira Forma Normal (3FN)

As regras referente à 3FN tem como objetivo solucionar um outro tipo de redundância, conforme apresentado pelo esquema abaixo:

– **2FN:**

Empregados (CódEmpregado, Nome, Categoria, Salário)

Detalhando o esquema apresentado anteriormente, note que a coluna Salário é determinada pela coluna Categoria. Nessa caso, salário é pago para uma determinada categoria, ora armazenada diversas vezes independentemente do número de empregados vinculados à mesma categoria.

A Figura 5.9 exemplifica esse tipo de problema, ora aplicada na tabela Empregados.

Tabela: Empregados			
<u>Cód Empregado</u>	Nome	Categoria	Salário
6124	Victor Nunes	A1	2
5134	Fábio Cardoso	A2	2
4211	José Ribeiro	C1	4
6162	Carlos da Silva	A2	2
1189	Sonia Purcini	A1	2

1189	Sonia Purcini	A1	2
4211	José Ribeiro	C1	2
2141	João de Freitas	A2	4

Figura 5.9 – Tabela “Empregados”

Na sequência, analisaremos as dependências funcionais existentes no esquema que representa a tabela Empregados.



A seta de cor azul escura representa o que denominamos de dependência funcional transitiva.

Uma tabela se encontra na 3FN, se e somente se, além de atender as regras da 2FN e não possuir dependências funcionais transitivas. Para que a tabela Empregados atenda as regras da 3FN, torna-se necessário eliminar a dependência funcional transitiva presente nela, deixando apenas colunas que dependem da chave-primária na tabela original, conforme demonstrado pelo esquema a seguir:

– **2FN:**



– **3FN:**

Empregados (CódEmpregado, Nome, Categoría)

Dessa maneira, as colunas que dependem de uma coluna que não é chave-primária devem constituir uma nova tabela, conforme apresentado pelo esquema abaixo, onde criamos uma nova tabela intitulada de Categoria com os atributos CódCategoria e Salário.

– **3FN:**

Categorias (CódCategoria, Salário)

O esquema final, após contemplar as regras da 3FN será constituído pelas seguintes tabelas a seguir:

– **3FN:**

Projetos (CódProjeto, Tipo, Descrição)

Projetos_Empregados (CódProjeto, CódEmpregado, DataInício, Tempo)

Empregados (CódEmpregado, Nome, Categoria)

Finalizando o processo de normalização que atende a 3FN, apresentamos o esquema anterior constituído pelas tabelas Projetos, Empregados, Projetos_Empregados e Categorias com suas respectivas instâncias, conforme podemos visualizar na Figura 5.10.

<u>CódProjeto</u>	Tipo	Descrição
PSI001	Desenvolvimento de Software	Sistema ERP
PSI034	Manutenção	Sistema Apoio à Decisão

Tabela: Projetos_empregados

<u>CódProjeto</u>	<u>CódEmpregado</u>	Data início	Tempo
PIS001	6124	11/01/2001	48
PIS001	5134	10/02/2001	48
PIS001	4211	10/03/2002	36

PIS001	6162	10/04/2002	36
PIS001	1189	11/01/2002	24
PSI034	1189	05/01/2003	24
PSI034	4211	01/04/2001	48
PSI034	2141	11/01/2002	24

Tabela: Empregados

<u>Cód Empregado</u>	Nome	Categoria
6124	Victor Nunes	A1
5134	Fábio Cardoso	A2
4211	José Ribeiro	C1
6162	Carlos daSilva	A2
1189	Sonia Purcini	A1
1189	Sonia Purcini	A1
4211	José Ribeiro	C1
2141	João de Freitas	A2

Tabela: Categorias

CódCategoría	Salários
A1	2
A2	2
C1	4

Figura 5.10 – Tabelas resultantes do esquema (3FN).

5.1.5 Regra: Quarta Forma Normal (4FN)

Basicamente, para a grande maioria dos bancos de dados, aplicar regras de normalização que atenda até a 3FN já é considerado suficiente. Normalmente, torna-se necessário decompor ainda mais o banco de dados, implementando outras formais normais, como, a citar a 4FN.

Considere o relacionamento de grau três apresentado pela Figura 5.11 a seguir, constituído pelas entidades PROJETO, EQUIPAMENTO e EMPREGADO, as quais se relacionam por intermédio do relacionamento ora identificado de UTILIZAÇÃO.

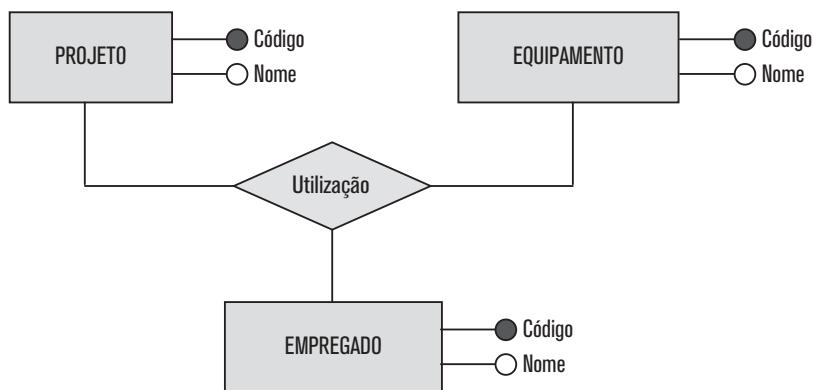


Figura 5.11 – Relacionamento Ternário

Suponha que seja necessário realizar o controle dos equipamentos utilizados nos projetos, como também, os empregados alocados aos projetos. Um possível mapeamento para o relacionamento UTILIZAÇÃO seria similar ao apresentado abaixo:

Utilização (CódProjeto, CódEmpregado, CódEquipamento)

Vamos considerar esses dados para a tabela UTILIZAÇÃO, conforme apresentado na Figura 5.12.

Tabela: Utilização

<u>CódProjeto</u>	<u>CódEmpregado</u>	<u>CódEquipamento</u>
1	1	1
1	2	1
1	3	1
1	1	2
1	2	2
1	3	2
2	2	2
2	2	4
3	3	1
3	4	1
3	3	3
3	4	3
3	3	5
3	4	5
4	2	5

Figura 5.12 – Instâncias da Tabela “Utilização”

Agora que possuímos a tabela UTILIZAÇÃO com seus respectivos dados, você consegue identificar quais são os empregados alocados no projeto cujo seu código corresponda ao número 1? Achou estranho? Existe duplicidade de dados na tabela? Para piorar ainda mais a situação, tente responder quais são os equipamentos utilizados no projeto de número 1?

Não se preocupe, estamos lidando com um tipo particular de dependência funcional, a chamada de multivalorada. A Figura 5.13 apresenta exemplos des-

se tipo de dependência funcional. Note que a sua notação é diferente das demais dependências funcionais estudadas até o presente momento.

Tabela: Utilização		
CódProjeto	CódEmpregado	CódEquipamento
1	1	1
1	2	1
1	3	1
1	1	2
	:	
	:	

$\text{CódProjeto} \rightarrow\!\!\! \rightarrow \text{CódEmpregado}$

Tabela: Utilização		
CódProjeto	CódEmpregado	CódEquipamento
1	1	1
1	2	1
1	3	1
1	1	2
	:	
	:	

$\text{CódProjeto} \rightarrow\!\!\! \rightarrow \text{CódEquipamento}$

Figura 5.13 – Dependência Funcional Multivalorada

Para que uma tabela atenda as regras da 4FN, além de atender as regras da 3FN, não poderá existir mais de uma dependência funcional multivalorada. Para adequarmos a relação UTILIZAÇÃO para a 4FN, devemos realizar a seguinte modificação:

- 3FN:

Utilização (CódProjeto, CódEmpregado, CódEquipamento)



- 4FN:

Projetos_Empregados (CódProjeto, CódEmpregado)

Projetos_Equipamentos (CódProjeto, CódEquipamento)

Dessa forma, podemos concluir que as tabelas Projetos_Empregados e Projetos_Equipamentos presentes no esquema anterior contemplam as regras pertinentes a 4FN.

5.1.6 Regra: Quinta Forma Normal (5FN)

Uma tabela se encontra na 5FN se a mesma não possuir nenhuma dependência funcional cíclica. Mas afinal, o que significa uma dependência funcional cíclica?

Uma dependência funcional cíclica ocorre quando as seguintes dependências estão presentes em uma tabela:

$$A \rightarrow B; B \rightarrow C; C \rightarrow A$$

Ou seja, B depende funcionalmente de A, C depende funcionalmente de B e, por sua vez, A depende funcionalmente de C, fechando um ciclo, caracterizando uma dependência funcional cíclica.



ATENÇÃO

Comparando as regras da 3FN e BCNF, é factível de identificar a existência de vantagens para a 3FN, na medida em que sabemos que sempre é possível obter um projeto que atenda as regras da 3FN sem sacrificar a preservação de dependência ou da ausência de perda. (Silberschatz, et Al, 2006).

Para ilustrar melhor a ocorrência de uma dependência funcional cíclica, suponha os requisitos de negócio de uma instituição de ensino superior, onde um professor pode ministrar diversas disciplinas (Professor Disciplina), isso é, nesse caso, dizemos que Disciplina depende funcionalmente de Professor, conforme demonstrada pela tabela PROFESSOR ora representada pela Figura 5.14.

Tabela: Professor	
Professor	Disciplina
José Ribeiro	Banco de Dados 1
	Bancos de Dados 2
Fernando Feliciano	Engenharia de Software
	Análise de Projeto de Sistemas

Ainda existe a possibilidade de um ou vários professores escreverem uma ou diversas apostilas, formando assim a dependência funcional cíclica, conforme podemos constatar na tabela Professor_Apostila representada pela Figura 5.15.

Tabela: Professor_Apostila

Professor	Apostila
Prof. Dr. Marcos Felipe	
Prof. Ms. Hélio da Silva	Banco de Dados 1 e 2
Prof. Dr. Joaquim dos Santos	
Prof. Ms. Mário da Cruz	Gerenciamento de Projetos
Prof. Esp. José Cardoso	

Figura 5.15 – Dependência Funcional Cíclica (Professor_Apostila).

Por fim, a exemplificação da dependência funcional cíclica, a tabela Disciplina_Professor é utilizada para representar que cada disciplina pode possuir diversas apostilas (Disciplina → Apostila), conforme ilustrado pela Figura 5.16.

Tabela: Disciplina_Apostila

Disciplina	Apostila
Banco de Dados 1	
Bancos de Dados 2	Banco de Dados 1 e 2
Engenharia de Software	Engenharia de Software
	Análise e Projeto de Sistemas

Figura 5.16 – Dependência Funcional Cíclica (Disciplina_Apostila).

Posteriormente a visualização das três tabelas (Professor, Professor_Apostila e Disciplina_Apostila), é possível identificar a relação cíclica existente entre Professor → Disciplina; Disciplina → Apostila e Professor → Apostila.



CONEXÃO

Leia mais sobre normalização de dados em: <<http://imasters.com.br/artigo/7020/banco-de-dados/modelagem-de-dados-final-normalizacao/>>

No processo de normalização de bancos de dados relacionais, eventualmente, podemos nos deparar com alguns problemas, por exemplo, o uso de chaves primárias incorretas ou até mesmo omitidas; atributos considerados relevantes não representados corretamente e possíveis atributos não relevantes, redundantes ou derivados.

Chaves-primárias omitidas ou utilizadas de forma inadequada podem surgir em uma tabela devido ao conceito de uma chave-primária não ser obrigatória, dessa forma, existe a possibilidade de encontrarmos tabelas onde a chave-primária se encontra ausente.

Caso você se deparar com uma tabela que não possui uma chave-primária ou, se a chave-primária existente nessa tabela seja distinta da chave-primária usual, para o processo de normalização de dados, deve-se considerar como se a chave-primária estivesse presente e, principalmente, inseri-la na forma não normalizada.



ATIVIDADE

5. Conceitue adequadamente o processo de normalização de dados. Cite suas principais formas.
6. Defina com suas palavras o conceito de dependência funcional parcial.
7. Apresente um exemplo de dependência funcional transitiva.
8. Defina adequadamente as dependências funcionais existentes em uma tabela que não contempla as regras da 4FN e 5FN.



REFLEXÃO

Parabéns! Você está se tornando um especialista em modelagem de dados!

Neste capítulo estudamos as principais regras utilizadas para realizar a normalização de da-

dos. A normalização de dados é um processo trivial em um projeto de banco de dados. Por meio da normalização eliminamos as anomalias de inserção, alteração e remoção.

Além de compreendermos os principais tipos de dependências funcionais, agora seremos capazes de identificá-las e ao mesmo tempo, eliminá-las a fim de estruturarmos adequadamente nossa relação.

Vimos também que a utilização das regras pertinentes a 4FN e 5FN torna-se necessário apenas em esquemas de banco de dados bem específicos, sobretudo por não ser habitual encontrarmos dependências funcionais multivaloradas e cíclicas.



LEITURA

Artigos on-line: para você incrementar mais o seu nível de aprendizagem do processo de normalização de dados:

<http://www.devmedia.com.br/normalizacao-de-banco-de-dados/29555>

<http://www.devmedia.com.br/passo-a-passo-para-modelagem-de-dados/28326>

Livro: Sistema de Banco de Dados, Silberschatz, A., Korth, H. F., Sudarshan, S. Neste livro você encontrará um conteúdo específico de normalização de dados, muito desta unidade foi baseada nos conceitos deste livro. Vale a pena sua consulta. Dedique-se bastante sobre o processo de normalização de dados, conhecer as principais regras de normalização é de grande importância para manipularmos dados de forma íntegra e concisa.



REFERÊNCIAS BIBLIOGRÁFICAS

ELMASRI, R.; NAVATHE, S.B. Sistemas de bancos de dados. São Paulo: Pearson (Addison Wesley), 2005.

KORTH, H.; SILBERCHATZ, A. Sistemas de bancos de dados. 3. ed. São Paulo: Makron Books, 1998.

HEUSER, C. A. Projeto de Bancos de Dados. 2. ed. Porto Alegre: Sagra Luzzatto, 1999.

PRESSMAN, R.S. Engenharia de software. São Paulo: Makron Books, 1995.

ROB, P.; CORONEL, C. Sistemas de Banco de Dados: Projeto, Implementação e Administração. 8. Ed. São Paulo: Cengage Learning, 2011.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de Banco de Dados. 5. ed. Rio de Janeiro: Elsevier, 2006.

DATE, C. J.; Introdução a Sistemas de Banco de Dados; 8^a. Ed.; Trad. Daniel Vieira; Rio de Janeiro: Elsevier, 2003.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4 ed. Instituto de Informática da UFRGS, Sagra DC Luzzatto, 1998.

RAMAKRISHNAN, R. GEHRKE, J. Database Management Systems. 2. ed. Boston: McGraw-Hill, 2000.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T.; JAGADISH, H. V.; Database Modeling and Design – Logical Design. 5^a ed., Burlington – USA: Elsevier, 2011.



EXERCÍCIO RESOLVIDO

Capítulo 1

1. Cite pelo menos três exemplos de Dados e Informações, num contexto de empresarial qualquer. Detalhe cada um.

Sugestão de Resposta: Podemos exemplificar dados e informações da seguinte maneira. Imagine respectivamente que você encontre esses três dados em um e-mail existente em sua caixa de mensagens: R\$ 2.567,00, Rafael Carvalho e Elizabeth Mendes da Costa. Bem, você pode concordar que esses dados não evidenciam nenhuma informação, por simplesmente desconhecermos o contexto o qual os mesmos estão inseridos. Agora, suponha que o primeiro dado (R\$ 2.567,00) esteja associado ao contexto de uma conta bancária, assim, poderíamos subtender que esse dado agora seria a informação do saldo atual dessa mesma conta. Por sua vez, o segundo dado se encontra incluso no contexto acadêmico, sendo assim, Rafael Carvalho se trataria do nome do aluno que foi aprovado em uma determinada disciplina. E, por fim, o dado Elizabeth Mendes da Costa, esteja associado ao contexto hospitalar, gerando a informação de que esse dado diz respeito

ao nome de uma paciente qualquer. Dessa maneira, podemos concluir de que o dado único e exclusivamente sozinho não nos gera nenhuma informação, tornando-se imprescindível conhecermos o contexto (regra de negócio) o qual o mesmo se encontra associado.

2. Apresente quatro diferenças significativas existentes entre um sistema de arquivo e um SGBD.

Sugestão de Resposta: Se compararmos o armazenamento de dados utilizando-se um sistema de arquivo e um SGBD, alguns problemas podem ser evidenciados quando utilizamos a primeira opção, a citar: a estrutura de arquivos é definida pelo próprio código-fonte do sistema computacional, prejudicando consideravelmente sua manutenção; o controle de acesso desses arquivos apresentam grandes obstáculos quando mencionamos o compartilhamento; o uso de formatos específicos acarreta no isolamento de dados e por fim, a ausência de controle de acesso concorrente pode gerar inconsistências nos dados. Já o armazenamento de dados por meio do uso de um SGBD elimina esses problemas reportados anteriormente, e ainda, implementa o uso de uma linguagem padrão (SQL) para promover adequadamente o acesso (DCL) e manipulação dos dados (DML) e objetos (DDL) de um banco de dados.

3. Identifique cinco características de um sistema de gerenciamento de banco de dados (SGBD).

Sugestão de Resposta: Evidenciamos como 5 principais características associadas a um SGBD o: controle de acesso aos dados; controle de redundância; compartilhamento de dados com controle de concorrência; múltiplas visões do mesmo conjunto de dados e restrições de integridade, seja a nível de entidade e ou referencial.

Capítulo 2

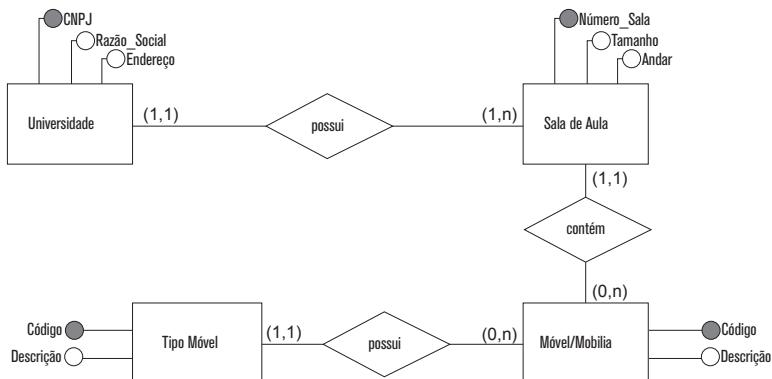
1. Descreva detalhadamente o conceito de Entidade e Relacionamento. Cite pelo menos três exemplos onde podemos utilizar ambos.

Sugestão de Resposta: Entidade pode ser considerada como algo que desejamos armazenar no banco de dados, a citar como exemplos: um carro, um funcionário e ou um aluno, que, por sua vez, representa um determinado tipo de objeto abstruído do mundo real. Um relacionamento tem como propósito descrever um vínculo (associação) entre uma ou vá-

rias entidades. Como primeiro exemplo, suponha a existência de um relacionamento o qual associa as entidades “Funcionário” e “Cliente” essa nomeada de “atende”, que pode ser interpretado da seguinte maneira: um funcionário atende um ou vários clientes. Um cliente é atendido por nenhuma, ou no máximo um funcionário. Para o segundo exemplo, considere um relacionamento nomeado de “trabalha” que associa as entidades “Departamento” e “Funcionário”. A interpretação desse relacionamento seria algo similar a: em um departamento pode trabalhar nenhum ou vários funcionários, por outro lado, um funcionário pode trabalhar em no máximo um departamento. E para finalizar os exemplos de relacionamentos, suponha a existência das entidades “Professor” e “Disciplina”, um relacionamento sugestivo seria “ministra”. Sua interpretação poderia ser algo como: um professor ministra um ou várias disciplinas, e pode sua vez, uma disciplina pode ser ministrada por um ou vários professores.

- Analise o cenário do ambiente acadêmico, mais especificamente, de uma sala de aula. A partir dessa analise, represente por meio de um DER, o conjunto de carteiras e o conjunto dos tipos de móveis.

Sugestão de Resposta:



- Discorra sobre os detalhes pertinentes ao Modelo Hierárquico, apontando suas desvantagens comparando com o Modelo em Rede.

Sugestão de Resposta: A estrutura lógica do modelo hierárquico é constituída por uma estrutura similar a estrutura de uma árvore, essa visualizada de cima para baixo, permitindo a

visualização das suas respectivas ramificações. Como principais desvantagens, comparando-se com o modelo de rede, podemos citar: representação exclusiva do relacionamento um para muitos (1:M) existente entre o segmento pai e seus respectivos filhos, isto é, cada segmento pai possui diversos segmentos filhos, porém, cada segmento filho, por sua vez, possui apenas vinculado a ele um segmento pai; ausência de independência estrutural e a dificuldade em gerenciar e manipular registros.

4. Realize uma pesquisa na Internet e descreva três características de um Banco de Dados XML. Cite pelo menos três nomes de banco de dados que manipulam arquivos XML.
Sugestão de Resposta: Oracle, PostgreSQL e SQL Server.
5. Dê pelo menos três exemplos de restrição aplicada a um modelo de dados. Na sequência, descreva os três tipos de relacionamentos que podem ser utilizado para associar entidades.
Sugestão de Resposta: Exemplos de restrições aplicadas a um modelo de dados qualquer: restrição de chaveprimária, restrição de chave-estrangeira e restrição de unicidade. Os três tipos de relacionamentos possíveis para associar entidades seriam: uma para muitos (1:M ou 1..*), muitos para muitos (M:N ou *..*) e um para um (1:1 ou 1..1).

Capítulo 3

1. Conceitue adequadamente um atributo, e discorra sobre os seus principais tipos. Na sequência, dê pelo menos um exemplo de cada tipo.
Sugestão de Resposta: um atributo é conceituado como uma característica particular de uma entidade, ou até mesmo de um relacionamento específico. Os atributos podem ser segmentados em diversos tipos, a citar: simples , composto, multivalorado, derivado e identificador.
Exemplos: Simples: nome; Composto: endereço; Multivalorado: telefone; Derivado: idade e Identificador: CPF.
2. Conceitue um relacionamento e classifique os relacionamentos em relação ao número de objetos envolvidos.
Sugestão de Resposta: Um relacionamento tem como propósito descrever um vínculo (associação) entre uma ou várias entidades. O grau de um relacionamento é determinado

pelo número de entidades participantes do mesmo relacionamento. A citar unário (grau um), binário (grau dois), ternário (grau três) e "n"ário (faz uso de mais de três entidades).

- Imagine um contexto acadêmico, o qual, poderíamos considerar uma sala de aula. Qual seria a cardinalidade máxima de um professor em relação aos alunos, como também, dos alunos em relação ao professor?

Sugestão de Resposta:



Capítulo 4

- Conceitue chave-primária simples e composta. Dê pelo menos três exemplo de cada tipo de chave-primária.

Sugestão de Resposta: uma chave-primária possui algumas características relevantes, a citar: os atributos definidos para constituir a chave-primária, por definição, têm que possuir valores únicos para cada registro na relação; nenhum dos atributos que constituem a chave-primária poderá, em hipótese alguma, possuir valores nulos em nenhum registro e no caso da chave-primária ser composta, não poderá ser adicionado mais atributos do que os mínimos necessários para identificar os registros de forma unívoca. Uma chave-primária simples necessita de apenas um atributo para identificar exclusivamente uma ocorrência de uma entidade qualquer, todavia, uma chave-primária composta, precisa de mais de um atributo para identificar uma ocorrência de endidade.

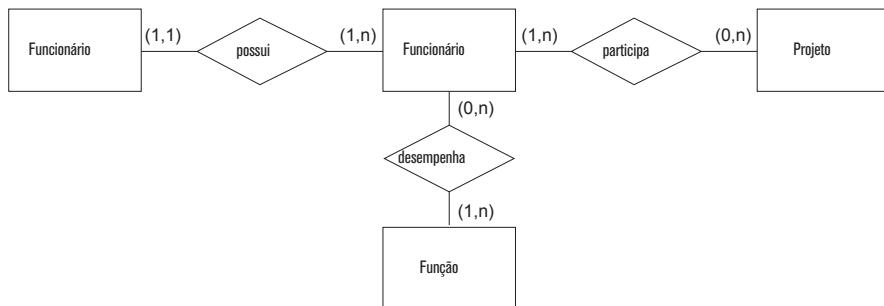
- Utilize suas próprias palavras para discorrer sobre os conceitos de Restrição de Integridade de Entidade (RIE) e Restrição de Integridade Referencial (RIR). Cite exemplos para ambos os conceitos.

Sugestão de Resposta: A restrição de integridade de entidade (RIE) tem como propósito garantir o acesso aos dados sem nenhuma ambiguidade. Para exemplificar o emprego da RIE, considere uma tupla qualquer, ora existente na relação R, dizemos que o valor de cada atributo que constitui a chave-primária de (t) deve ser diferente de nulo e, ainda, não poderá haver outra tupla (t') na relação (R) com o mesmo

valor de chave-primária de (t) . Já a restrição de integridade referencial, poderá ser exemplificada considerando uma tupla (t) qualquer e um chave-estrangeira em (t) , o valor de chave-estrangeira pode ser nulo se e somente se os atributos de chave-estrangeira não formarem a chave-primária de (t) , e ainda, o valor da chave-estrangeira poderá ser diferente de nulo apenas se existir uma tupla (t) na relação referenciada tal que a chave-primária de (t) possuir o mesmo valor da chave-estrangeira de (t) .

- Analise os requisitos a seguir e constitua sua modelagem relacional que atenda algumas necessidades de informação, a citar: Qual o código e a descrição de cada projeto desempenhado na empresa? Qual é o número da matrícula e nome de cada funcionário? Quais são as possíveis funções desempenhadas na empresa?

Sugestão de Resposta:



- Imagine que uma empresa os funcionários trabalham em projetos, onde em cada projeto um funcionário poderá exercer diversas funções de acordo com as regras expostas abaixo:
 - Os funcionários podem realizar distintas funções em diversos projetos;
 - Eventualmente, um funcionário pode exercer em um mesmo projeto distintas funções;
 - Em um determinado projeto podemos ter uma mesma função (atribuição) exercida por distintos funcionários;
 - Por outro lado, um funcionário poderá realizar a mesma função em distintos projetos.

Sugestão de Resposta: (similar ao DER anterior, porém, agora incrementamos as cardinalidades a fim de atender as regras).

5. Quais características são desejáveis para uma chave-candidata?

Sugestão de Resposta: Uma chave-candidata é uma chave que apresenta obrigatoriamente as duas características a seguir: (1) unicidade: não há duas linhas (tuplas) distintas na tabela com o mesmo valor para os atributos da chave; (2) irredutibilidade: não existe um subconjunto de atributos da chave que apresentem a característica de unicidade.

Capítulo 5

1. Conceitue adequadamente o processo de normalização de dados. Cite suas principais formas

Sugestão de Resposta: O processo de normalização tem como objetivo promover a reestruturação dos dados a fim de eliminar qualquer tipo de redundância caracterizada como indesejável, ora presentes em alguns esquemas de banco de dados. Esse mesmo processo é constituído por cinco formas normais (1FN, 2FN, 3FN, 4FN e 5FN), entretanto, se for considerado as três primeiras formas normais (1FN, 2FN e 3FN), já se é possível alcançarmos satisfatoriamente um modelo de banco de dados estruturado e conciso.

2. Defina com suas palavras o conceito de dependência funcional parcial.

Sugestão de Resposta: Uma dependência funcional parcial é quando um determinado atributo (coluna) depende de apenas parte de uma chave-primária composta, ou seja, não depende da chave-primária composta inteira para identificar exclusivamente uma ocorrência dessa entidade (relação/tabela).

3. Apresente um exemplo de dependência funcional transitiva.

Sugestão de Resposta: Uma dependência funcional transitiva é quando um atributo (coluna) depende de outro atributo (coluna) que não é a chave-primária e nem faz parte de uma chave-primária composta para identificar uma ocorrência de entidade (relação/tabela) exclusivamente.

Exemplo: Considere a relação abaixo intitulada de “Funcionários”:

Funcionários (CódFuncionário, Nome, Sobrenome, Endereço, Função, Salário)

Os valores do atributo nomeado de “Salário” depende dos valores do atributo “Função” e não da chave-primária “CódFuncionário” para identificar exclusivamente uma ocorrência de entidade (relação/tabela), caracterizando o que chamamos de dependência funcional transitiva.

4. Define adequadamente as dependências funcionais existentes em uma tabela que não contempla as regras da 4FN e 5FN.

Sugestão de Resposta: Uma tabela para contemplar as regras pertinentes a 4FN e 5FN devem respectivamente eliminar o que denominamos de dependência funcional multivalorada e transitiva
