

FUNDAMENTOS DE BANCOS DE DADOS



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Reitor

ANTONIO JOSÉ DE ALMEIDA MEIRELLES

Coordenadora Geral da Universidade

MARIA LUIZA MORETTI



Conselho Editorial

Presidente

EDWIGES MARIA MORATO

ALEXANDRE DA SILVA SIMÕES – CARLOS RAUL ETULAIN
CICERO ROMÃO RESENDE DE ARAUJO – DIRCE DJANIRA PACHECO E ZAN
IARA BELELI – IARA LIS SCHIAVINATTO – MARCO AURÉLIO CREMASCO
PEDRO CUNHA DE HOLANDA – SÁVIO MACHADO CAVALCANTE

Célio Cardoso Guimarães

FUNDAMENTOS DE BANCOS DE DADOS
Modelagem, projeto e linguagem SQL

FICHA CATALOGRÁFICA ELABORADA PELO
SISTEMA DE BIBLIOTECAS DA UNICAMP
DIVISÃO DE TRATAMENTO DA INFORMAÇÃO

G947f Guimarães, Célio Cardoso, 1942-
Fundamentos de bancos de dados : modelagem, projeto e linguagem SQL /
Célio Cardoso Guimarães. – Campinas, SP : Editora da Unicamp, 2003.

1. Banco de dados. 2. Modelagem de dados. 3. SQL (Linguagem de programação de computador). I. Título.

ISBN 85-268-0633-5 CDD 001.6
005.7565

Índices para catálogo sistemático:

| | |
|-------------------------------------------------|----------|
| 1. Banco de dados | 001.6 |
| 2. Modelagem de dados | 001.6 |
| 3. SQL (Linguagem de programação de computador) | 005.7565 |

Copyright © by Célio Cardoso Guimarães
Copyright © 2003 by Editora da Unicamp

5ª reimpressão, 2023

Opiniões, hipóteses e conclusões ou recomendações expressas
neste livro são de responsabilidade do autor e não
necessariamente refletem a visão da Editora da Unicamp.

Direitos reservados e protegidos pela lei 9.610 de 19.2.1998.
É proibida a reprodução total ou parcial sem autorização,
por escrito, dos detentores dos direitos.

Foi feito o depósito legal.

Direitos reservados a

Editora da Unicamp
Rua Sérgio Buarque de Holanda, 421 – 3ª andar
Campus Unicamp
CEP 13083-859 – Campinas – SP – Brasil
Tel./Fax: (19) 3521-7718 / 7728
www.editoraunicamp.com.br – vendas@editora.unicamp.br

Títulos em Engenharia de Software

A série Títulos em Engenharia de Software tem por objetivo cobrir a grande maioria dos tópicos abordados na área de engenharia de software. Os livros da série foram escritos por professores do Instituto de Computação da Universidade Estadual de Campinas, que ensinam ou trabalham nessa área, em alguns casos em parceria com professores de outras universidades brasileiras. A motivação para a produção da série vem da carência de textos didáticos, em português, sobre esses assuntos. Ademais, como se trata de uma série, existe um comprometimento entre os vários assuntos abordados pelos diversos textos, de forma que eles sejam complementares. O público a que a série se destina são os alunos de graduação e pós-graduação e profissionais da área de desenvolvimento de produtos de software.

O primeiro livro faz uma introdução aos conceitos que serão tratados com mais profundidade nos outros livros. O texto apresenta as principais atividades envolvidas na engenharia de software, utilizando exemplos práticos para mostrar os diferentes aspectos do processo de construção de software, desde a sua concepção até a sua instalação. O livro é indicado especialmente para pessoas com formação em computação, mas sem conhecimento da área e que desejem adquirir uma visão ampla dos vários assuntos envolvidos em engenharia de software. Deve ser visto como ponto de partida para a escolha de tópicos específicos para aprofundamento posterior. Todos os outros livros da série visam introduzir os conceitos básicos utilizados na construção de produtos de software, tais como metodologias para desenvolvimento de sistemas, qualidade de software e gerenciamento de projetos.

Os livros que tratam de metodologias de desenvolvimento discutem os conceitos envolvidos nas atividades de construção de produtos de software, utilizando uma das seguintes abordagens: estruturada, orientada a objetos, orientada a dados, formal ou orientada a sistemas interativos, aplicadas a casos reais. O desenvolvimento estruturado é apresentado através de conceitos básicos e de ferramentas utilizadas na análise e projeto estruturados, tais como diagrama de fluxo de dados, diagrama de fluxo de eventos, dicionário de dados e diagrama de estrutura de software. A especificação do projeto a partir da análise estruturada e a qualidade do projeto são também discutidas. O desenvolvimento orientado a objetos é apresentado através da construção

do modelo de objetos, diagramas de seqüência de mensagens, diagramas de colaboração entre objetos e casos de uso. Abordagens formais para o desenvolvimento de sistemas são discutidas através da apresentação de técnicas para construção de especificações que possibilitem uma verificação rigorosa das propriedades da especificação construída. Os fundamentos necessários para um perfeito entendimento da notação Z, linguagem largamente empregada em projetos industriais, assim como um material mais avançado sobre ela, são apresentados. O desenvolvimento de sistemas voltado ao projeto de dados é apresentado através de bases de dados relacionais, utilizando exemplos na linguagem SQL. O texto inclui modelagem, projeto lógico, conceitos de normalização, de integridade e de transações. O desenvolvimento de sistemas interativos é apresentado através das questões envolvidas nas tarefas típicas da construção de interfaces homem-computador (IHC). Além disso, um panorama das diferentes abordagens para desenvolvimento de interfaces é apresentado.

Os livros voltados para a qualidade de software destinam-se a discutir a qualidade do processo e do produto, através de modelos e técnicas para garantia de qualidade. Os modelos de qualidade são apresentados a partir dos conceitos básicos e do relacionamento entre os modelos. São analisados os principais modelos de qualidade de software, desde aqueles que se concentram no produto até a tendência mais atual de modelos centrados na qualidade do processo. São também discutidos os desdobramentos mais recentes ocorridos nos modelos CMMI, ISO 9000:2000 e ISO 15504. As técnicas para garantia de qualidade são abordadas através da aplicação de procedimentos tais como inspeção, passeio e testes, tanto para a abordagem estruturada como para a abordagem orientada a objetos. São também apresentados aspectos referentes à manutenção de software, tais como processo de manutenção, reengenharia de software e ferramentas CASE.

O livro dedicado ao gerenciamento de projetos de software discute a importância das métricas para as atividades envolvidas na gerência de projetos. Nele são apresentadas diversas métricas úteis para o gerenciamento e sua utilização nas estimativas de custo, prazo e na organização de projetos. Além disso, é discutido o papel das métricas na tomada de decisão e análise de risco, atividades envolvidas em projetos de software.

All times I have enjoy'd
Greatly, have suffer'd greatly, both with those
That loved me, and alone.
I am a part of all that I have met.

To strive, to seek, to find, and not to yield.

ALFRED TENNYSON, *Ulysses*

Prefácio

Este livro é o resultado da compilação de notas de aula ministradas pelo autor nos últimos oito anos em cursos introdutórios de bancos de dados para alunos de Engenharia de Computação e Ciência da Computação da UNICAMP e em cursos da Escola de Extensão em Engenharia de Software do Instituto de Computação da UNICAMP.

Apesar da existência de ótimos textos introdutórios na área de bancos de dados, como [Ullman97] e [Elsmani94], acreditamos que este livro vem preencher uma lacuna de textos em língua portuguesa pelo seu enfoque no modelo relacional, predominante nos atuais sistemas comerciais e de código aberto de bancos de dados, e na linguagem SQL, incluindo recursos recentes dos padrões SQL/92 e SQL:1999.

O texto é voltado para estudantes que queiram ter uma visão unificada de modelagem de dados, projeto de bancos de dados relacionais e programação em SQL, e para praticantes e administradores da área de bancos de dados que queiram ter uma visão coerente desses três tópicos.

O texto está organizado da seguinte forma: o capítulo 1 dá uma introdução a características marcantes de sistemas gerenciadores de bancos de dados e aos mecanismos de comunicação de estações clientes com servidores de bancos de dados. O capítulo 2 introduz técnicas de modelagem de dados centradas no modelo entidade-relacionamento. O capítulo 3 trata do projeto lógico de dados focado no modelo relacional; as principais características do modelo relacional são aí apresentadas. O capítulo 4 descreve as metalinguagens de manipulação de dados relacionais, álgebra relacional e cálculo relacional de tuplas. O capítulo 5 apresenta os conceitos básicos de dependências funcionais e de normalização (“formas normais”); é o mais pesado do ponto de vista teórico. O capítulo 6 contém uma introdução abrangente à linguagem SQL, começando com características básicas e terminando com técnicas e exemplos de consultas mais avançadas. Ele depende apenas do capítulo 3 e pode ser lido independentemente dos demais. Os conceitos e sintaxe da linguagem SQL são apresentados informalmente por meio de exemplos, o que julgamos ser a forma mais eficaz para se aprender uma nova linguagem. Restringimo-nos aos recursos do padrão SQL/92, evitando descrever recursos “adicionais” de fornecedores, exceto em alguns casos especiais. O capítulo 7

apresenta recursos avançados do padrão SQL: restrições de integridade, asserções, procedimentos armazenados, e “triggers”. O capítulo 8 apresenta técnicas para programação em SQL através de uma linguagem hospedeira, incluindo um exemplo de programação com SQL dinâmico através de uma biblioteca independente de SGBD. O capítulo 9 apresenta de forma resumida os conceitos de segurança, transações e controle de concorrência. O capítulo 10 descreve alguns recursos importantes do padrão SQL:1999: novos tipos de dados, recursão em consultas SQL, e orientação a objetos. O apêndice apresenta uma comparação simples de vários SGBDs, incluindo exemplos de programação com SQL dinâmico para vários deles.

Os pré-requisitos para este texto são conhecimentos básicos de técnicas de programação em alguma linguagem algorítmica como C, Pascal etc. Eles aparecem de forma subjacente nas linguagens apresentadas nos capítulos 4 e 6. Conhecimentos elementares de teoria de conjuntos e lógica ajudarão a compreensão dos capítulos 4, 5 e 6. Tópicos mais avançados ou especializados e que podem ser deixados de lado numa primeira leitura estão marcados com um asterisco ‘*’ e tópicos mais complexos com dois ‘**’ asteriscos.

Textos introdutórios sobre bancos de dados têm a sorte de poder utilizar exemplos concisos para ilustrar a maioria dos conceitos básicos. Além dos exemplos e exercícios que criamos, não quisemos “reinventar a roda”, e recorremos e adaptamos exemplos e exercícios de vários livros da área, em particular [Ullman97], [Ullman88], [Date97] e [Setzer89].

Cada capítulo vem com vários exercícios que ajudarão o leitor a sedimentar e aprofundar os conceitos apresentados. Recomendamos que pelo menos alguns exercícios de cada capítulo sejam trabalhados. Exercícios mais difíceis estão precedidos com um ou dois asteriscos. No apêndice são também descritos ambientes SQL de uso livre para escolas e universidades, onde o poder da linguagem SQL pode ser plenamente explorado. Um portal para o livro foi criado em <http://www.ic.unicamp.br/~celio/livrobd>, onde se encontram *scripts* para criar e preencher a base de dados exemplo em qualquer SGBD relacional, assim como o texto das consultas e exemplos SQL apresentados no capítulo 6.

Agradecimentos

As seguintes pessoas e entidades contribuíram para a realização deste trabalho, a quem gostaríamos de expressar nossos sinceros agradecimentos: professora Thelma Chiossi, pela oportunidade de apresentá-lo como disciplina da Escola de Extensão em Engenharia de Software, e pelo incansável empenho na publicação deste livro; professor Nelson Machado, pelo encorajamento contínuo, pela cessão da biblioteca *ps_pack.c*, utilizada na construção das figuras do texto, e pelas sugestões que muito contribuíram para me-

lhorar a sua qualidade; professores Ricardo Nery e Claudia Medeiros, pelas sugestões e revisão de parte do texto; professores Arnaldo Moura, Ariadne Carvalho, Jorge Stolfi e Tomasz Kowaltowski, pela ajuda com o ambiente \LaTeX , a Editora da Unicamp pela presteza e suporte na publicação deste livro-texto, e o Instituto de Computação da Unicamp, pela infra-estrutura, suporte computacional e, principalmente, pelo excelente ambiente de trabalho.

Célio Guimarães
Campinas, julho de 2003

Sumário

| | | |
|----------|--------------------------------------------------------------------------------------------------|-----------|
| 1 | Conceitos básicos | 19 |
| 1.1 | Bancos de dados <i>versus</i> coleção de arquivos | 20 |
| 1.2 | Sistemas gerenciadores de bases de dados | 21 |
| 1.3 | Um exemplo: a base de dados <i>Torneios de Tênis da ATP</i> . . . | 23 |
| 1.4 | Conexões lógicas a sistemas gerenciadores de bases de dados . | 25 |
| 1.4.1 | Conexão tipo <i>time sharing</i> | 25 |
| 1.4.2 | Conexão tipo <i>servidor de arquivos</i> | 26 |
| 1.4.3 | Conexão tipo <i>cliente-servidor</i> | 26 |
| 1.4.4 | Conexão via <i>servidor de aplicações</i> | 28 |
| 1.4.5 | Conexão via <i>servidor Web</i> (Internet) | 28 |
| 2 | Modelagem de dados | 31 |
| 2.1 | Projeto <i>top down</i> de bases de dados | 31 |
| 2.2 | O modelo entidade-relacionamento | 33 |
| 2.2.1 | Tipos de atributos | 35 |
| 2.2.2 | Relacionamentos | 36 |
| 2.2.3 | Conjuntos de entidades fracas | 39 |
| 2.2.4 | Auto-relacionamentos | 39 |
| 2.2.5 | Extensões ao modelo entidade-relacionamento | 41 |
| 2.2.6 | Um exemplo de projeto conceitual usando o MER: a BD <i>Torneios de Tênis da ATP</i> | 44 |
| 2.2.7 | Convenções para nomes — considerações práticas . . . | 46 |
| 2.2.8 | Exercícios | 47 |
| 3 | Projeto lógico de dados | 51 |
| 3.1 | Modelos lógicos históricos | 51 |
| 3.2 | O modelo relacional | 52 |
| 3.2.1 | Valores nulos | 55 |
| 3.2.2 | Restrições de integridade | 55 |
| 3.2.3 | Restrições sobre atributos | 56 |
| 3.2.4 | Mapeamento do MER para o modelo relacional | 57 |

| | | |
|----------|-------------------------------------------------------------------------------------|-----------|
| 3.2.5 | Mapeamento do projeto conceitual da BD <i>Torneios de Tênis</i> para o MR | 61 |
| 3.2.6 | Exercícios | 61 |
| 4 | Linguagens de manipulação de dados relacionais | 63 |
| 4.1 | Álgebra relacional (AR) | 63 |
| 4.1.1 | Exemplos de consultas usando álgebra relacional | 67 |
| 4.1.2 | Um exemplo mais elaborado: composições de materiais | 69 |
| 4.1.3 | * Operador divisão da álgebra relacional | 71 |
| 4.2 | Cálculo relacional de tuplas (CRT) | 73 |
| 4.2.1 | Exercícios resolvidos sobre CRT | 76 |
| 4.3 | Exercícios | 77 |
| 5 | Conceitos de normalização de dados relacionais | 81 |
| 5.1 | Motivação | 81 |
| 5.2 | O conceito de dependência funcional (df) | 82 |
| 5.2.1 | Regras de inferência de Armstrong | 83 |
| 5.2.2 | Formas normais (FN) de relações | 84 |
| 5.2.3 | Quarta forma normal | 86 |
| 5.2.4 | * O conceito de fecho de um conjunto de atributos | 88 |
| 5.2.5 | * Decomposição de relações não normalizadas | 90 |
| 5.3 | Considerações práticas sobre normalização | 94 |
| 5.4 | Exercícios | 95 |
| 6 | A linguagem SQL | 99 |
| 6.1 | Histórico | 99 |
| 6.2 | Características básicas | 100 |
| 6.2.1 | O ambiente de um SGBD relacional | 101 |
| 6.2.2 | Tipos de dados | 103 |
| 6.2.3 | Comandos para definição de dados | 104 |
| 6.2.4 | Comandos para alterar a definição de tabelas | 109 |
| 6.2.5 | Um guia prático para nomes de tabelas e colunas | 110 |
| 6.3 | O comando <i>select</i> | 110 |
| 6.3.1 | Características básicas | 111 |
| 6.3.2 | Operadores de comparação | 113 |
| 6.3.3 | Conectores lógicos <i>and</i> , <i>or</i> , <i>not</i> | 114 |
| 6.3.4 | Operador <i>between</i> | 114 |
| 6.3.5 | Operadores <i>in</i> , <i>not in</i> | 114 |
| 6.3.6 | Operador <i>like</i> | 115 |
| 6.3.7 | Funções para manipular cadeias de caracteres | 116 |
| 6.3.8 | Operação de junção | 118 |
| 6.3.9 | Sinônimos para nomes de tabelas (<i>alias</i>) | 119 |
| 6.3.10 | Problemas com operandos nulos | 120 |

| | | |
|--------|------------------------------------------------------------------------------------------------------------|-----|
| 6.3.11 | Subconsultas com o operador <i>in</i> | 122 |
| 6.3.12 | Subconsultas com operador de comparação | 123 |
| 6.3.13 | Operadores <i>all</i> e <i>any</i> em subconsultas | 124 |
| 6.3.14 | Operadores <i>exists</i> e <i>not exists</i> | 125 |
| 6.3.15 | Escopo de nomes em comandos <i>select</i> | 126 |
| 6.3.16 | Operadores união (<i>union</i>), interseção (<i>intersect</i>) e diferença (<i>except</i>) | 127 |
| 6.3.17 | Cláusula <i>order by</i> | 128 |
| 6.3.18 | Funções que produzem valores agregados | 129 |
| 6.3.19 | Funções agregado na cláusula <i>select</i> | 129 |
| 6.3.20 | Miscelânea | 131 |
| 6.4 | Atualização de tabelas | 132 |
| 6.4.1 | Comando <i>insert</i> | 132 |
| 6.4.2 | Comando <i>update</i> | 133 |
| 6.4.3 | Comando <i>delete</i> | 134 |
| 6.4.4 | Tabelas temporárias, tabelas criadas dinamicamente | 134 |
| 6.5 | * Características avançadas do comando <i>select</i> | 135 |
| 6.5.1 | Cláusula <i>group by</i> | 135 |
| 6.5.2 | Cláusula <i>select</i> com função agregado seguida de cláusula <i>group by</i> | 136 |
| 6.5.3 | Cláusula <i>having</i> | 137 |
| 6.5.4 | Exercícios | 139 |
| 6.5.5 | Junção natural e junção exterior | 144 |
| 6.6 | Visões (<i>Views</i>) | 146 |
| 6.6.1 | Comandos para manipular visões | 147 |
| 6.6.2 | Atualização de visões | 149 |
| 6.7 | * Expressões com tipos de dados para datas e horas | 150 |
| 6.7.1 | Visão geral | 151 |
| 6.7.2 | Expressões aritméticas envolvendo <i>datetime</i> e <i>interval</i> | 153 |
| 6.7.3 | Exemplos | 154 |
| 6.7.4 | Compatibilidade de atribuições e comparações | 155 |
| 6.8 | * Usos avançados do comando <i>select</i> | 156 |
| 6.8.1 | Exercícios | 162 |
| 6.8.2 | ** Expressões-tabela | 163 |
| 6.8.3 | Exercícios | 171 |
| 6.8.4 | ** Utilização dos operadores de diferença e interseção na formulação de consultas complexas | 174 |
| 6.8.5 | Um roteiro para elaboração de consultas complexas | 176 |
| 6.8.6 | Exercícios | 177 |
| 6.8.7 | Operadores <i>case</i> , <i>match</i> e <i>unique</i> | 177 |
| 6.8.8 | Idiosincrasias, peculiaridades | 179 |
| 6.8.9 | Exercícios | 180 |

| | | |
|-----------|---------------------------------------------------------------------------------------|------------|
| 7 | Restrições de integridade, <i>stored procedures</i> e <i>triggers</i> | 185 |
| 7.1 | Restrições de integridade | 185 |
| 7.1.1 | Restrições sobre uma tabela base | 186 |
| 7.1.2 | Restrições sobre tabela base do tipo <i>check (condition)</i> | 187 |
| 7.1.3 | Restrições sobre domínios | 189 |
| 7.1.4 | Asserções | 190 |
| 7.1.5 | Tratamento de violação de restrições de integridade referencial | 191 |
| 7.1.6 | Verificação postergada de restrições | 192 |
| 7.1.7 | Exercícios | 193 |
| 7.2 | Procedimentos armazenados (<i>stored procedures</i>) | 194 |
| 7.3 | SQL procedural | 196 |
| 7.4 | Gatilhos (<i>triggers</i>) | 199 |
| 7.4.1 | Visão geral | 199 |
| 7.4.2 | Quando usar <i>before triggers</i> ou <i>after triggers</i> ? | 203 |
| 7.4.3 | Problemas na implementação de gatilhos | 203 |
| 7.4.4 | Exercícios | 204 |
| 8 | Programação SQL via linguagem hospedeira | 205 |
| 8.1 | Sintaxe básica | 205 |
| 8.1.1 | SQL dinâmico | 210 |
| 8.1.2 | Um exemplo: SQL dinâmico para acesso a BDs relacionais heterogêneas via Web | 212 |
| 9 | Administração e segurança de dados, transações | 217 |
| 9.1 | Organização de um SGBD padrão SQL | 217 |
| 9.1.1 | Tabelas do catálogo (<i>information_schema</i>) | 218 |
| 9.2 | Segurança de dados | 219 |
| 9.3 | O conceito de transação | 222 |
| 9.3.1 | Transações concorrentes | 225 |
| 9.3.2 | Controle de concorrência | 226 |
| 9.3.3 | Níveis de isolamento de transações em SQL/92 | 228 |
| 10 | Recursos do padrão SQL:1999 | 231 |
| 10.1 | Extensões ao modelo relacional | 231 |
| 10.1.1 | Novos tipos de dados | 231 |
| 10.1.2 | Novos predicados | 232 |
| 10.1.3 | Operadores <i>Rollup</i> e <i>Cube</i> | 233 |
| 10.1.4 | Privilégios para <i>roles</i> | 235 |
| 10.2 | ** Recursão em consultas SQL | 235 |
| 10.2.1 | Exercícios | 238 |
| 10.3 | Recursos de orientação a objetos | 240 |

| | | |
|----------|---------------------------------------------------------------|------------|
| A | Comparação de SGBDs relacionais | 243 |
| A.1 | Exemplos de acesso a bases de dados relacionais via Php . . . | 244 |
| A.1.1 | Exemplo de acesso a uma BD MySql | 244 |
| A.1.2 | Exemplo de acesso a uma BD Firebird ou Interbase . . | 247 |
| A.1.3 | Exemplo de acesso a uma BD Postgres | 249 |
| A.1.4 | Exemplo de acesso a uma BD Oracle | 250 |
| A.2 | Um exemplo completo: processador SQL interativo | 253 |
| A.3 | Ambientes de programação SQL | 260 |
| A.3.1 | Arquivos de configuração | 262 |
| | Referências bibliográficas | 263 |
| | Índice remissivo | 265 |

Capítulo 1

Conceitos básicos

Um *banco de dados* ou *base de dados* (*BD*) é uma coleção de dados ou informações relacionadas entre si. Elas representam aspectos do mundo real com significado próprio e que desejamos armazenar para uso futuro. Usaremos os dois termos acima de forma intercambiável, isto é, com o mesmo significado.

Ao se projetar um banco de dados tem-se em mente um conjunto de aplicações que primordialmente se deseja fazer sobre os dados. Elas determinam o uso principal que se quer fazer do banco de dados. Bancos de dados podem ser muito simples ou muito complexos e/ou de tamanhos que podem variar ordens de magnitude de um banco de dados para outro. Uma base de dados pequena e simples, por exemplo, poderia ser a relação dos nomes e telefones das pessoas conhecidas de um indivíduo ou dos bens e valores de uma pessoa física. Bases de dados podem ser mantidas manualmente, como as duas citadas acima, ou num computador. Apenas estas nos interessam neste estudo. Na verdade, o barateamento espetacular dos microcomputadores tornou econômico e eficaz manter em computador bases de dados antes mantidas manualmente, como o estoque de uma pequena farmácia ou de uma padaria. Uma base de dados grande poderia ser, por exemplo, o catálogo de todos os livros publicados nos Estados Unidos ou uma base de dados de todas as fotos recolhidas ao longo dos anos pelo programa espacial americano. Esta última poderia ocupar centenas de terabytes (1 terabyte $\approx 10^{12}$ bytes) e, provavelmente, não poderia ser armazenada *on-line* por nenhum computador moderno.

Bases de dados complexas envolvem muitos tipos diferentes de dados interdependentes e inter-relacionados como os dados corporativos de uma grande organização como a Boeing ou a General Motors, podendo chegar a centenas ou milhares de tipos distintos de dados: dados sobre funcionários, sobre fábricas, sobre produtos, sobre matérias-primas, sobre componentes, sobre finanças etc. Cada tipo de dados de uma base de dados corporativa

pode conter milhares ou até milhões de itens, como o número de funcionários ou o número de componentes que entram na fabricação dos aviões da Boeing ou dos automóveis da GM. Muitas vezes o nome que se dá à base de dados nos dá uma idéia do uso principal que se pretende fazer da mesma. Exemplos:

- Sistema de reserva de passagens aéreas
- Base de dados de biblioteca
- Base de dados dos contribuintes da Receita Federal
- Base de dados acadêmicos da UNICAMP
- Base de dados de uma cadeia de distribuição de bens de consumo (*Retail Store*)
- Sistema bancário do banco BXY
- Base de dados da rede de cabos telefônicos da Companhia Telefônica

Até recentemente bases de dados costumavam armazenar unicamente dados alfanuméricos (tipicamente cadeias de caracteres e valores numéricos). Atualmente elas estão armazenando também imagens, gráficos, e até objetos multimídia (som e vídeo), o que aumenta enormemente as necessidades de armazenamento e a complexidade de recuperação e processamento dos dados. A última aplicação acima, por exemplo, poderia incluir informações geográficas da rede de cabos da Companhia Telefônica, e seria um caso típico de uma nova especialização de bancos de dados, os chamados *bancos de dados geográficos*.

1.1 Bancos de dados *versus* coleção de arquivos

Na definição acima de bancos de dados enfatizamos a expressão: *dados relacionados entre si*. Esta característica fundamental de bancos de dados modernos distingue-os de uma conceituação mais antiga, que seria a de uma coleção de arquivos tradicionais suportados pelo sistema operacional. Tal coleção não reflete o inter-relacionamento que existe entre os dados nem as regras de consistência que explicitam esses inter-relacionamentos e que serão descritas adiante. Essas regras podem ser muito simples como, por exemplo, requerer que a lista dos funcionários que fazem parte de um departamento esteja contida na lista global de funcionários mantida pelo setor de RH e também na lista mantida pelo setor de folha de pagamento e contabilidade. Outras regras de consistência podem ser bastante complexas envolvendo, por