

# Sistemas de Gerenciamento de Banco de Dados

Tradução da  
Terceira Edição



Mc  
Graw  
Hill

Ramakrishnan • Gehrke

## Sistemas de Gerenciamento de Banco de Dados

ISBN 978-85-7726-027-0

A reprodução total ou parcial deste volume por quaisquer formas ou meios, sem o consentimento escrito da editora, é ilegal e configura apropriação indevida dos direitos intelectuais e patrimoniais dos autores.

### Copyright © 2008 de McGraw-Hill Interamericana do Brasil Ltda.

Todos os direitos reservados.

Av. Brigadeiro Faria Lima, 201 – 17º. andar

São Paulo, SP, CEP 05426-100

Todos os direitos reservados. Copyright © 2008 de McGraw-Hill Interamericana Editores, S. A. de C. V.

Prol. Paseo de la Reforma 1015 Torre A Piso 17, Col. Desarrollo Santa Fé, Delegación Alvaro Obregón

México 01376, D. F., México

Tradução da terceira edição do original em inglês Database Management Systems.

© 2003, 2000, 1998 de The McGraw-Hill Companies, Inc.

ISBN da obra original: 0-07-246563-8

Diretor-Geral: *Adilson Pereira*

Editora: *Gisélia Costa*

Supervisora de Produção: *Guacira Simonelli*

Preparação de Texto: *Lucrécia Freitas e Mônica de Aguiar*

Design da Capa: *Mick Wiggins*

Editoração Eletrônica: *Crontec Ltda.*

---

R165s      Ramarkrishnan, Raghu.  
                 Sistemas de gerenciamento de banco de dados  
                 [recurso eletrônico] / Raghu Ramarkrishnan, Johannes  
                 Gehrke ; tradução: Célia Taniwake. – 3. ed. – Dados  
                 eletrônicos. – Porto Alegre : AMGH, 2011.

Editado também como livro impresso em 2008.

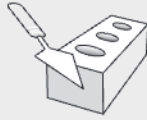
ISBN 978-85-63308-77-1

1. Ciência da computação. 2. Bases de dados –  
Gerenciamento. I. Gehrke, Johannes. II. Título.

CDU 004.658

---

Catlogação na publicação: Ana Paula Magnus – CRB 10/2052



# 1

---

## VISÃO GERAL SOBRE SISTEMAS DE BANCO DE DADOS

- ☛ O que é um SGBD, em particular, um SGBD relacional?
- ☛ Por que devemos utilizar um SGBD para gerenciar dados?
- ☛ Como os dados da aplicação são representados em um SGBD?
- ☛ Como os dados em um SGBD são recuperados e manipulados?
- ☛ Como um SGBD suporta o acesso concorrente e protege os dados na ocorrência de falhas no sistema?
- ☛ Quais são os principais componentes de um SGBD?
- ☛ Quem está envolvido com bancos de dados na vida real?
- **Conceitos-chave:** gerenciamento de banco de dados, independência de dados, projeto de banco de dados, modelo de dados; bancos de dados e consultas relacionais; esquemas, níveis de abstração; transações, concorrência e bloqueio, recuperação e registro em log; arquitetura de um SGBD; administrador de um banco de dados, programador do aplicativo, usuário final.

Reparou que todas as letras da palavra *database* (banco de dados, em inglês) são digitadas com a mão esquerda? Sabemos que a disposição do teclado da máquina de escrever (QWERTY) foi projetada, entre outras coisas, para facilitar o uso uniforme de ambas as mãos. Conclui-se, então, que escrever sobre bancos de dados, além de ser algo não natural, é bem mais difícil do que parece.

— Anônimo

A quantidade de informações que nos são disponíveis está literalmente explodindo, e o valor dos dados como um ativo organizacional é amplamente reconhecido. Para obter a maior parte de seus grandes e complexos conjuntos de dados, os usuários necessitam de ferramentas que simplifiquem as tarefas de gerenciamento dos dados e a extração de informações úteis de forma oportuna. Caso contrário, os dados podem se tornar

A área de sistemas de gerenciamento de dados é um microcosmo da Ciência da Computação em geral. Os aspectos tratados e as técnicas utilizadas abrangem um amplo espectro, incluindo linguagens, orientação a objeto e outros paradigmas de programação, compilação, sistemas operacionais, programação concorrente, estruturas de dados, algoritmos, teoria, sistemas distribuídos e paralelos, interfaces do usuário, sistemas especialistas e inteligência artificial, técnicas estatísticas e programação dinâmica. Não podemos tratar todos esses aspectos de gerenciamento de banco de dados em um livro, mas esperamos prover ao leitor um sentido de investigação nesta disciplina rica e vibrante.

um passivo, cujo custo de aquisição e gerenciamento excede em muito o valor por ele produzido.

Um **banco de dados** é uma coleção de dados que, tipicamente, descreve as atividades de uma ou mais organizações relacionadas. Por exemplo, um banco de dados de uma universidade poderia conter informações sobre:

- *Entidades* como alunos, professores, cursos e turmas.
- *Relacionamentos* entre as entidades, como a matrícula dos alunos nos cursos, cursos ministrados pelos professores, e o uso de salas por cursos.

Um **sistema de gerenciamento de banco de dados**, ou **SGBD**, é um software projetado para auxiliar a manutenção e utilização de vastos conjuntos de dados. A necessidade de tais sistemas, assim como seu uso, tem crescido rapidamente. A alternativa para não se usar um SGBD é armazenar os dados em arquivos e escrever código específico do aplicativo para gerenciá-los. O uso de um SGBD tem diversas vantagens importantes, como veremos na Seção 1.4.

## 1.1 GERENCIANDO DADOS

O objetivo deste livro é apresentar uma introdução detalhada dos sistemas de gerenciamento de banco de dados, com ênfase em como *projetar* um banco de dados e como *usar* efetivamente um SGBD. Naturalmente, várias decisões sobre como utilizar um SGBD para um determinado aplicativo dependem de quais recursos o SGBD suporta de forma eficiente. Assim, para aproveitar bem um SGBD, é necessário compreender também como ele *funciona*.

Diversos tipos de sistemas de gerenciamento de banco de dados estão em uso, mas este livro concentra-se nos **sistemas de banco de dados relacionais (SGBDRs)**, que com certeza constituem o tipo dominante de SGBD nos dias atuais. As seguintes questões são tratadas nos capítulos principais deste livro:

1. **Projeto de Banco de Dados e Desenvolvimento de Aplicativo:** Como um usuário pode descrever uma empresa do mundo real (por exemplo, uma universidade) em termos dos dados armazenados em um SGBD? Que fatores devem ser considerados ao decidir sobre a forma de organização dos dados armazenados? Como podemos desenvolver aplicativos que dependem de um SGBD? (Capítulos 2, 3, 6, 7, 19, 20 e 21.)



2. **Análise dos Dados:** Como um usuário pode responder a dúvidas sobre a empresa formulando consultas sobre os dados do SGBD? (Capítulos 4 e 5.)<sup>1</sup>
3. **Concorrência e Robustez:** Como um SGBD permite que vários usuários acessem os dados de forma concorrente, e como ele protege os dados na ocorrência de falhas do sistema? (Capítulos 16, 17 e 18.)
4. **Eficiência e Escalabilidade:** Como um SGBD armazena grandes conjuntos de dados e responde a questões sobre esses dados de forma eficiente? (Capítulos 8, 9, 10, 11, 12, 13, 14 e 15.)

Os capítulos posteriores abrangem tópicos importantes que estão evoluindo rapidamente, como o gerenciamento de banco de dados distribuído e paralelo, armazenagem de dados e consultas complexas para apoio à decisão, mineração de dados, recuperação de banco de dados e informações, repositórios XML, banco de dados orientado a objetos, gerenciamento de dados espaciais, e extensões de SGBD orientado a regras.

No restante deste capítulo, introduziremos estes tópicos. Na Seção 1.2, começamos com uma breve história da área e uma discussão do papel do gerenciamento de banco de dados nos sistemas de informações modernos. Identificaremos, então, os benefícios de armazenar os dados em um SGBD em vez de em um sistema de arquivos, na Seção 1.3, e, na Seção 1.4, discutiremos as vantagens de usar um SGBD para gerenciar dados. Na Seção 1.5, apresentaremos como as informações sobre uma empresa devem ser organizadas e armazenadas em um SGBD. Um usuário provavelmente pensa sobre as informações num alto nível, que corresponde às entidades da organização e seus relacionamentos, enquanto o SGBD basicamente armazena os dados na forma de (vários e vários) bits. A lacuna existente entre como os usuários pensam sobre seus dados e como os dados são definitivamente armazenados é preenchida através de diversos *níveis de abstração* suportados pelo SGBD. Intuitivamente, um usuário pode começar descrevendo os dados em termos totalmente de alto nível, e depois melhorar a descrição considerando o armazenamento adicional e detalhes de representação conforme necessário.

Na Seção 1.6, consideraremos como os usuários podem recuperar os dados armazenados em um SGBD e a necessidade de técnicas para processar eficientemente respostas às consultas envolvendo tais dados. Na Seção 1.7, forneceremos uma visão geral sobre como um SGBD suporta o acesso concorrente aos dados por diversos usuários e como ele protege os dados na ocorrência de falhas do sistema.

Descreveremos, então, brevemente, a estrutura interna de um SGBD na Seção 1.8, e, na Seção 1.9, mencionaremos vários grupos de pessoas associadas com o desenvolvimento e uso de um SGBD.

## 1.2 UMA PERSPECTIVA HISTÓRICA

Desde os primeiros computadores, armazenar e manipular dados têm sido o principal foco dos aplicativos. O primeiro SGBD de propósito geral, projetado por Charles Bachman, na General Electric, no início da década de 1960, foi chamado Depósito de Dados Integrados (Integrated Data Store). Ele constituiu a base do *modelo de dados de rede*, que foi padronizado pela Conference on Data Systems Languages (CODASYL) e influenciou bastante os sistemas de banco de dados na década de 1960. Bachman foi o primeiro a ser contemplado pelo Prêmio Turing da ACM (o equivalente ao Prêmio Nobel na Ciência da Computação) pelo trabalho na área de banco de dados; ele recebeu o prêmio em 1973.

---

<sup>1</sup> Um capítulo on-line sobre Consulta por Exemplo (QBE — Query-by-Example) também está disponível. Veja mais informações no Prefácio.

No final da década de 1960, a IBM desenvolveu o SGBD Sistema de Gerenciamento de Informação (IMS — Information Management System), ainda usado atualmente em diversas instalações. O IMS constituiu a base da estrutura de representação alternativa de dados, chamada *modelo de dados hierárquico*. O sistema SABRE para reservas de passagens aéreas foi desenvolvido em conjunto pela American Airlines e pela IBM nessa mesma época, e permitiu que diversas pessoas acessassem os mesmos dados através de uma rede de computadores. Interessante observar que, atualmente, o mesmo sistema SABRE é utilizado para fornecer serviços populares de viagens baseados na Web, tais como o Travelocity.

Em 1970, Edgar Codd, do Laboratório de Pesquisa de San Jose, da IBM, propôs uma nova estrutura de representação de dados chamada *modelo de dados relacional*, que veio a ser um marco histórico no desenvolvimento de sistemas de banco de dados. Ele impulsionou o rápido desenvolvimento de vários SGBDs baseados no modelo relacional, juntamente com um rico conjunto de resultados teóricos que consolidaram firmemente a área. Codd ganhou o Prêmio Turing de 1981 pelo seu trabalho original. Os sistemas de banco de dados amadureceram como uma disciplina acadêmica, e a popularidade dos SGBDs relacionais alterou o cenário comercial. Seus benefícios foram amplamente reconhecidos, e o uso de SGBDs para gerenciar dados corporativos tornou-se uma prática padrão.

Na década de 1980, o modelo relacional consolidou sua posição como o paradigma dominante de SGBD, e o uso dos sistemas de banco de dados continuou a se difundir cada vez mais. A linguagem de consulta SQL para banco de dados relacional, desenvolvida como parte do projeto System R, da IBM, passou a ser a linguagem de consulta padrão. A SQL foi padronizada no final dos anos 80, e o padrão atual, SQL:1999 foi adotado pelo American National Standards Institute (ANSI) e pela International Organization for Standardization (ISO). É possível argumentar que a forma mais amplamente utilizada de programação concorrente é a execução concorrente de programas de banco de dados (chamados *transações*). Os usuários escrevem os programas como se eles fossem executar isoladamente, e a responsabilidade de executá-los de forma concorrente é atribuída ao SGBD. James Gray ganhou o Prêmio Turing de 1999 pelas suas contribuições ao gerenciamento de transações de banco de dados.

No final da década de 1980 e na década de 1990, houve avanços em diversas áreas dos sistemas de banco de dados. Pesquisas consideráveis foram conduzidas para desenvolver linguagens de consulta mais poderosas e modelos de dados mais ricos, com ênfase no suporte à análise complexa de dados provenientes de todas as áreas da empresa. Diversos fabricantes (como o DB2 da IBM, Oracle 8, Informix<sup>2</sup> UDS) estenderam seus sistemas com a capacidade de armazenar novos tipos de dados, como imagens e texto, e a possibilidade de consultas mais complexas. Sistemas especializados têm sido desenvolvidos por inúmeros fabricantes para criação de *data warehouses*, consolidando os dados de diversos bancos de dados, com o intuito de conduzir análise especializada.

Um fenômeno interessante é a emergência de diversos pacotes de **planejamento de recurso empresarial (ERP — enterprise resource planning)** e de **planejamento de recurso de gerenciamento (MRP — management resource planning)**, que acrescentaram uma camada substancial de recursos orientados a aplicativos acima de um SGBD. Os pacotes largamente utilizados incluem sistemas da Baan, Oracle, PeopleSoft, SAP e Siebel. Esses pacotes identificam um conjunto de tarefas comuns (por exemplo, gerenciamento de inventário, planejamento de recursos humanos, análise financeira) desempenhadas por um grande número de organizações e fornecem uma camada de aplicativo genérica para realizar essas tarefas. O dado é armazenado em um SGBD

---

<sup>2</sup> A Informix foi recentemente adquirida pela IBM.



relacional, e a camada de aplicativo pode ser customizada para empresas diferentes, diminuindo os custos totais para as organizações, comparados ao custo de criar uma camada de aplicativo a partir do zero.

O fato histórico mais significativo, talvez, seja a entrada dos SGBDs na Era Internet. Enquanto a primeira geração de web sites armazenava seus dados exclusivamente em arquivos dos sistemas operacionais, o uso de um SGBD para armazenar dados acessados através de um navegador Web tem se difundido cada vez mais. As consultas são geradas através de formulários acessíveis na Web e as respostas são formatadas usando uma linguagem de marcação como o HTML para serem facilmente exibidas em um navegador. Todos os fabricantes de banco de dados estão acrescentando recursos aos seus SGBDs com o objetivo de torná-los mais adequados para desenvolvimento para Internet.

O gerenciamento de banco de dados continua a ganhar importância conforme mais e mais dados tornam-se disponíveis on-line e ainda mais acessíveis através da rede de computadores. Atualmente, a área está sendo impulsionada por ideais excitantes. Entre eles temos: banco de dados multimídia, vídeo interativo, fluxos de dados, bibliotecas digitais, um hospedeiro de projetos científicos, como o esforço de mapeamento do genoma humano, e o projeto de Sistema de Observação Terrestre da NASA, além do desejo das empresas de consolidar seus processos de tomada de decisão e *minerar* seus repositórios de dados por informações úteis sobre seus negócios. Comercialmente, os sistemas de gerenciamento de banco de dados representam um dos maiores e mais ativos segmentos de mercado. Assim, o estudo de sistemas de banco de dados pode vir a ser ricamente gratificante e não apenas de uma maneira, mas de várias!

### 1.3 ARQUIVOS DE SISTEMAS *VERSUS* UM SGBD

Para compreendermos a necessidade de um SGBD, consideremos um cenário motivador: uma empresa tem uma grande coleção (digamos 500 GB<sup>3</sup>) de dados sobre os funcionários, departamentos, produtos, vendas e assim por diante. Esse dado é acessado concorrentemente por diversos funcionários. As consultas sobre os dados devem ser respondidas rapidamente, as alterações realizadas nos dados pelos diferentes usuários devem ser aplicadas consistentemente, e o acesso a determinadas partes dos dados (por exemplo, salários) deve ser restrito.

Podemos experimentar gerenciar os dados armazenando-os em arquivos do sistema operacional. Essa abordagem tem várias desvantagens, que incluem:

- Provavelmente, não teremos 500 GB de memória principal para armazenar todos os dados. Devemos, então, armazenar os dados em um dispositivo de armazenamento, como disco ou fita, e carregar partes relevantes dos dados na memória principal conforme necessário.
- Mesmo que tivéssemos 500 GB de memória principal, num sistema computacional de 32 bits de endereçamento, não podemos nos referir diretamente a mais do que aproximadamente 4 GB de dados. Temos que programar algum método de identificação de todos os itens de dados.
- Devemos escrever programas especiais para responder a cada pergunta que um usuário pode desejar fazer sobre os dados. Esses programas provavelmente serão complexos em razão do grande volume de dados a ser pesquisado.

---

<sup>3</sup> Um kilobyte (KB) são 1024 bytes, um megabyte (MB) são 1024 KBs, um gigabyte (GB) são 1024 MBs, um terabyte (TB) são 1024 GBs, e um petabyte (PB) são 1024 terabytes.

- Devemos proteger os dados de alterações inconsistentes realizadas por usuários diferentes acessando os dados de forma concorrente. Se os aplicativos devem tratar dos detalhes desse acesso concorrente, isto aumenta significativamente a sua complexidade.
- Devemos assegurar que os dados sejam restaurados a um estado consistente se o sistema falhar enquanto as alterações estão sendo realizadas.
- Os sistemas operacionais provêm apenas um mecanismo de senha para segurança. Isso não é suficientemente flexível para reforçar as políticas de segurança nas quais usuários diferentes têm permissão de acessar subconjuntos diferentes dos dados.

Um SGBD é um pacote de software projetado para executar mais facilmente as tarefas mencionadas anteriormente. Armazenando-se dados em um SGBD em vez de em uma coleção de arquivos do sistema operacional, é possível utilizar os recursos do SGBD para gerenciar os dados de uma forma robusta e eficiente. À medida que cresce o volume de dados e o número de usuários — centenas de gigabytes de dados e milhares de usuários são comuns nos bancos de dados corporativos atuais —, o suporte de um SGBD torna-se indispensável.

## 1.4 VANTAGENS DE UM SGBD

Usar um SGBD para gerenciar dados tem várias vantagens:

- **Independência de Dados:** Os programas aplicativos não devem, idealmente, ser expostos aos detalhes de representação e armazenamento de dados. O SGBD provê uma visão abstrata dos dados que oculta tais detalhes.
- **Acesso Eficiente aos Dados:** Um SGBD utiliza uma variedade de técnicas sofisticadas para armazenar e recuperar dados eficientemente. Este recurso é especialmente importante se os dados são armazenados em dispositivos de armazenamento externos.
- **Integridade e Segurança dos Dados:** Se os dados são sempre acessados através do SGBD, ele pode forçar restrições de integridade. Por exemplo, antes de inserir informações sobre o salário de um funcionário, o SGBD pode verificar se o orçamento do departamento não está se excedendo. Além disso, ele pode forçar *controles de acesso* que governam quais dados estão visíveis a diferentes classes de usuários.
- **Administração de Dados:** Quando diversos usuários compartilham dados, centralizar a administração dos dados pode oferecer melhorias significativas. Profissionais experientes que compreendem a natureza dos dados sendo gerenciados, e como os diferentes grupos de usuários os utilizam, podem ser responsáveis por organizar a representação dos dados para minimizar a redundância e para realizar as sintonizações finas do armazenamento dos dados para garantir uma eficiente recuperação.
- **Acesso Concorrente e Recuperação de Falha:** Um SGBD planeja o acesso concorrente aos dados de maneira tal que os usuários podem achar que os dados estão sendo acessados por apenas um único usuário de cada vez. Além disso, o SGBD protege os usuários dos efeitos de falhas de sistema.
- **Tempo Reduzido de Desenvolvimento de Aplicativo:** Obviamente, o SGBD suporta funções importantes que são comuns a vários aplicativos que acessam os dados no SGBD. Isso, em conjunto com uma interface de alto nível aos dados, facilita o desenvolvimento rápido de aplicativos. Os aplicativos de SGBD tendem a ser



mais robustos do que os aplicativos similares independentes porque muitas tarefas importantes são tratadas pelo SGBD (e não precisam ser depuradas e testadas no aplicativo).

Dadas todas essas vantagens, há alguma razão para *não* se utilizar um SGBD? Algumas vezes, sim. Um SGBD é um software complexo, otimizado para alguns tipos de processamento (por exemplo, responder a consultas complexas ou tratar várias requisições concorrentes), e seu desempenho pode não ser adequado para determinados aplicativos especializados. Exemplos incluem aplicativos com restrições rígidas de tempo real ou algumas operações críticas bem definidas para as quais um código customizado eficiente deve ser escrito. Uma outra razão para não se utilizar um SGBD é que o aplicativo pode precisar manipular os dados de maneiras não suportadas pela linguagem de consulta. Em tais casos, a visualização abstrata dos dados apresentada pelo SGBD pode não corresponder às necessidades do aplicativo e realmente impossibilitar o seu uso. Como um exemplo, os bancos de dados relacionais não suportam a análise flexível de dados textuais (embora os fabricantes estejam atualmente estendendo seus produtos nesta direção).

Se o desempenho especializado ou solicitações de manipulação de dados são essenciais num aplicativo, pode-se optar por não utilizar um SGBD, especialmente se os benefícios de um SGBD (por exemplo, consulta flexível, segurança, acesso concorrente e recuperação de falha) não forem exigidos. Entretanto, na maioria das situações em que é necessário gerenciamento de dados em grande escala, os SGBDs têm se tornado uma ferramenta indispensável.

**Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.**