

Banco de Dados

VISÃO GERAL DE BANCOS DE DADOS

Introdução

O que é um banco de dados?

- Um banco de dados é uma coleção organizada de dados relacionados, na qual podemos realizar consultas a qualquer momento, permitindo, assim, extrair informações.
 - Os dados são fatos que podem ser armazenados e que tenham algum significado.
- Exemplo:
 - Nomes, números de telefones e endereços de pessoas que você conheça.
 - Esses dados podem estar armazenados em:
 - Uma agenda de papel indexada pelas letras do alfabeto;
 - ou em um arquivo de seu computador (por exemplo, uma planilha do Excel).
 - Essa coleção de dados relacionados com um significado implícito é um banco de dados.



Introdução

O que é um banco de dados?

- Os bancos de dados são componentes essenciais no nosso dia a dia.
 - Consulta ao acervo da biblioteca
 - Reserva em hotéis e linhas aéreas.
 - Saques, depósitos e transferências bancárias.
 - Compras on-line.
- Grande parte das informações que acessamos estão armazenadas em bancos de dados.
 - A maior parte dessas informações armazenadas são dados textuais ou numéricos.
 - No entanto, os bancos de dados também permitem o armazenamento de outros tipos de informação:
 - Imagens
 - Áudio e vídeo
 - Informações geográficas

Introdução

BD vs SGBD



- **Banco de Dados (BD)**

- Se refere aos dados armazenados e à estrutura e relacionamento entre esses dados.



- **Sistema de Gerenciamento de Bancos de Dados (SGBD)**

- Se refere ao sistema que implementa e gerencia diferentes bancos de dados, permitindo aos usuários consultarem e manipularem os dados.

Introdução

Propriedades de um banco de dados

- Um banco tem as seguintes propriedades:
 - **Representam algum contexto do mundo real**
 - Algumas vezes, esse contexto é denominado mini-mundo ou universo de discurso.
 - Mudanças nesse universo devem ser refletidas no banco de dados.
 - **É uma coleção lógica e coerente de dados com algum significado inerente**
 - Uma organização de dados ao acaso (aleatório) não pode ser corretamente interpretada como um banco de dados.
 - **Um banco de dados é projetado, construído e povoado por dados, atendendo a uma proposta específica**
 - Ele possui um grupo de usuários específico e aplicações preconcebidas, de acordo com o interesse desse grupo de usuários.

Introdução

Tamanho e complexidade dos bancos de dados

- Um banco de dados pode variar em tamanho e complexidade.
- **Lista de nomes e endereços**
 - Poucas centenas de registros de estruturas simples.
- **Catálogo computadorizado de uma biblioteca**
 - Pode conter milhares de registros organizados em diferentes categorias.
- **Banco de dados da Receita Federal**
 - Milhões de contribuintes e cada um deles fornecendo diversos tipos de dados.
 - Suponha que a Receita receba 25 milhões de declarações de imposto de renda.
 - Cada declaração com 300 caracteres (bytes de informação).
 - $25.000.000 \times 300 = 6.98 \text{ GiB}$ de informação

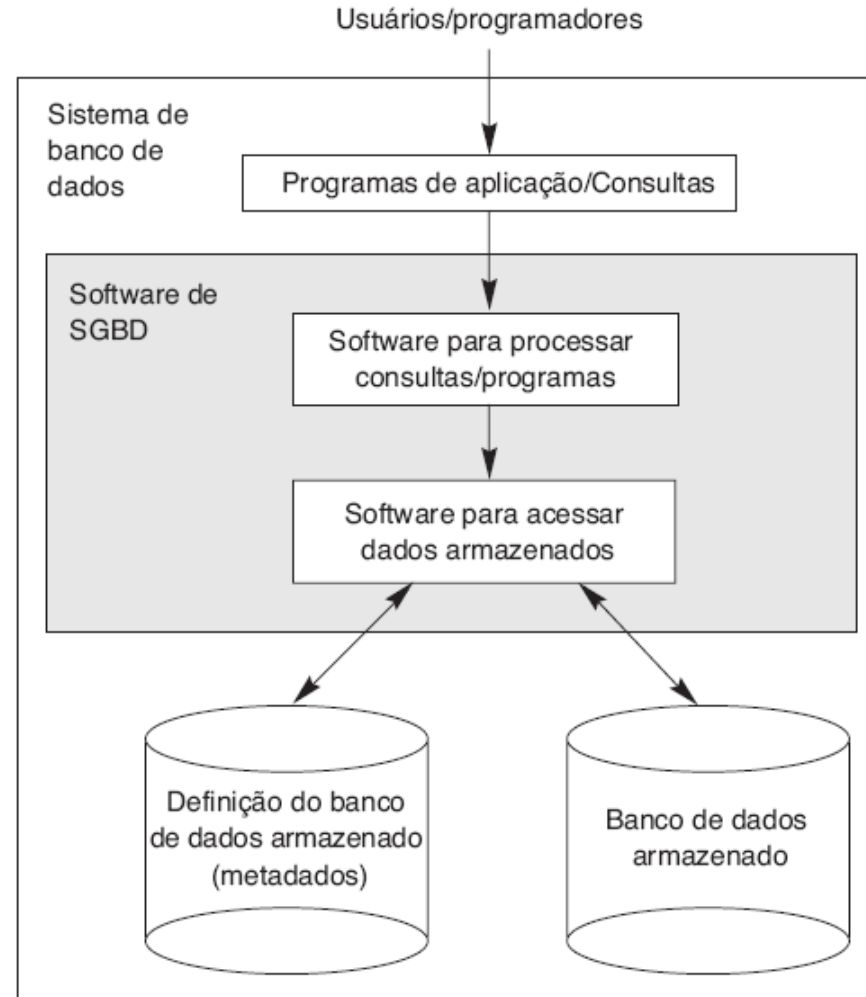
Introdução

Sistema de Gerenciamento de Bancos de Dados (SGBD)

- Um SGBD é uma coleção de programas que permite aos usuários criar e manter um ou mais bancos de dados. Dessa forma, um SGBD é sistema de software de propósito geral que facilita os seguintes processos:
 - **Definição de dados**
 - Especificar os tipos de dados, estruturas e restrições dos dados que serão armazenados no banco de dados.
 - **Construção**
 - É o processo de armazenar dados em alguma mídia de armazenamento que é controlada pelo SGBD.
 - **Manipulação**
 - Inclui funções como consultas ao banco de dados para recuperar dados específicos, atualizar os dados e gerar relatórios.
 - **Compartilhamento**
 - Permite o acesso de múltiplos usuários e aplicações a uma base de dados simultaneamente.
 - **Proteção**
 - Inclui proteção de falhas no sistema (mau funcionamento de hardware/software) e proteção de segurança (acessos não autorizados).
 - **Manutenção:**
 - Permitir que o sistema de banco de dados evolua de acordo com os requisitos de mudanças ao longo do tempo.

Introdução

Visão simplificada de um ambiente de banco de dados



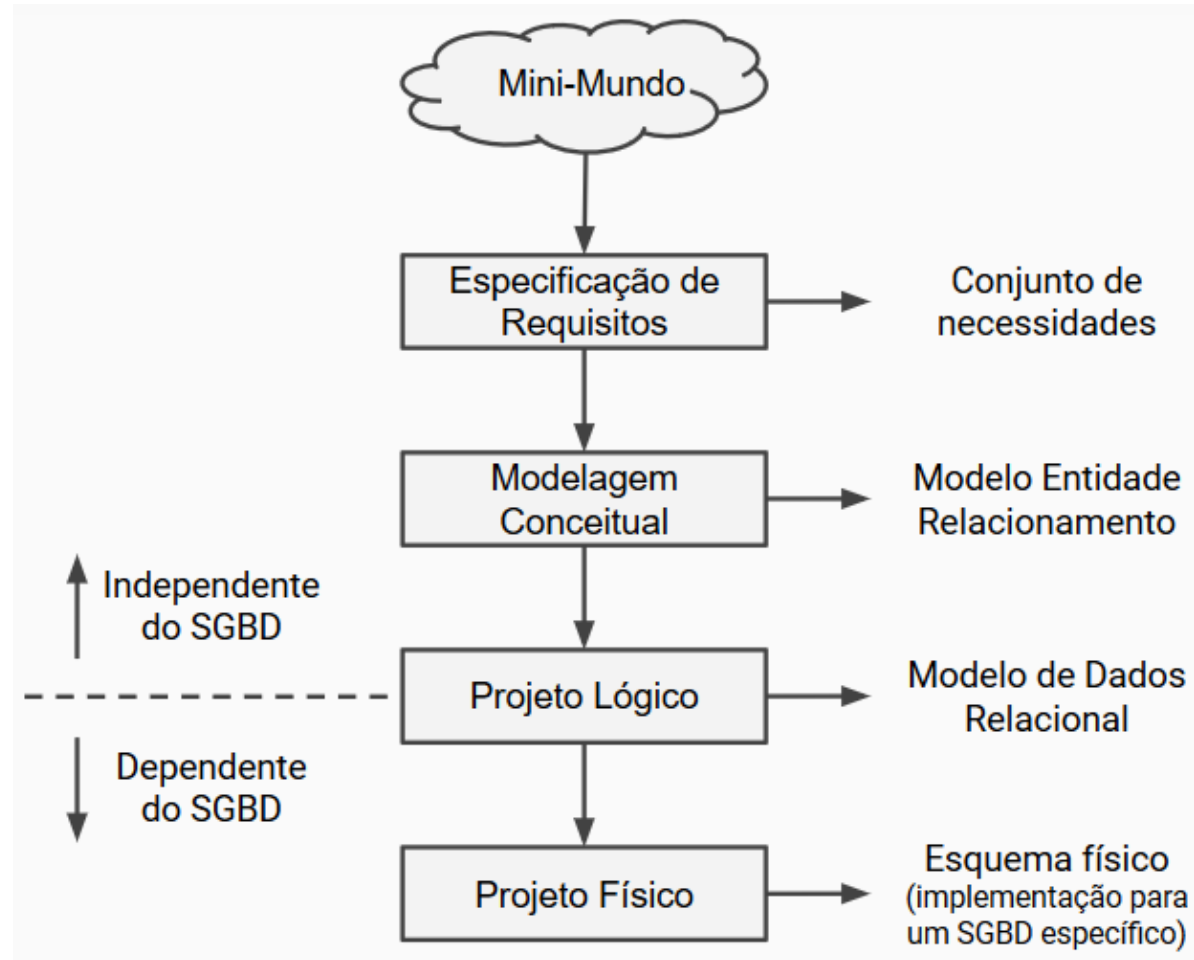
Introdução

Projeto de bancos de dados

- O projeto de um banco de dados passa por algumas fases:
 - **Especificação e análise de requisitos**
 - Nesta etapa são identificadas as necessidades do sistema que será desenvolvido.
 - **Modelagem conceitual**
 - Define um modelo de dados abstratos, que descreve a estrutura de um banco de dados de forma independente do SGBD.
 - Nela são registrados os dados que podem aparecer no banco de dados, mas não determina como esses dados serão armazenados a nível de SGBD.
 - O resultado desta etapa é um modelo Entidade-Relacionamento (ER), geralmente representado graficamente através de um Diagrama Entidade-Relacionamento (DER).
 - **Projeto lógico**
 - Define um modelo de dados que representa a estrutura de dados de um banco de dados conforme vista pelo usuário do SGBD.
 - É dependente do SGBD (relacional, orientado a objetos, objeto-relacional, NoSQL).
 - Em um banco de dados relacional, são definidas quais tabelas o banco de dados irá conter e, para cada tabela, quais as suas colunas.
 - **Projeto físico**
 - O modelo do banco de dados é enriquecido com detalhes que influenciam no desempenho do banco de dados, mas não interferem em sua funcionalidade.
 - Alterações nesse modelo não afetam as aplicações que usam o banco de dados.
 - Na prática, é um processo contínuo, que ocorre mesmo depois de o banco de dados já estar implementado e em funcionamento.
 - Por exemplo, a definição do formato de arquivo, índices, entre outros.

Introdução

Projeto de bancos de dados



SGBDs vs Arquivos

Assim como os arquivos simples de texto ou arquivos binários, os SGBDs possibilitam o armazenamento persistente de dados para os programas.

Quais as vantagens de uso dos SGBDs?

- Controle de redundância.
- Restringir acesso não autorizado.
- Prover estruturas de armazenamento e técnicas de busca para o processamento eficiente de consultas.
- Prover *backup* e restauração dos dados.
- Oferecer múltiplas interfaces de conexão aos dados.
- Representar relacionamentos complexos entre os dados.
- Garantir a integridade dos dados.

SGBDs vs Arquivos

Assim como os arquivos simples de texto ou arquivos binários, os SGBDs possibilitam o armazenamento persistente de dados para os programas.

Quais as vantagens de uso dos SGBDs?

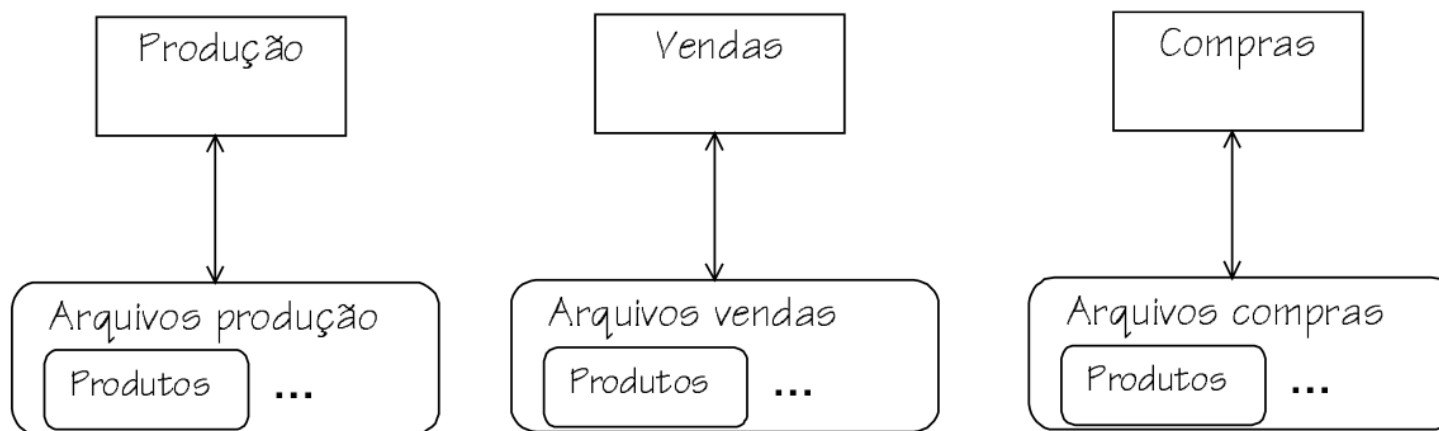
- Controle de redundância.
- Restringir acesso não autorizado.
- Prover estruturas de armazenamento e técnicas de busca para o processamento eficiente de consultas.
- Prover *backup* e restauração dos dados.
- Oferecer múltiplas interfaces de conexão aos dados.
- Representar relacionamentos complexos entre os dados.
- Garantir a integridade dos dados.



Apesar das vantagens possibilitadas por um SGBD, existem situações em que o uso de arquivos de texto simples são mais eficientes. Por exemplo, quando os dados utilizados quase nunca sofrem alterações e não possuem relacionamentos complexos.

Controle de redundância

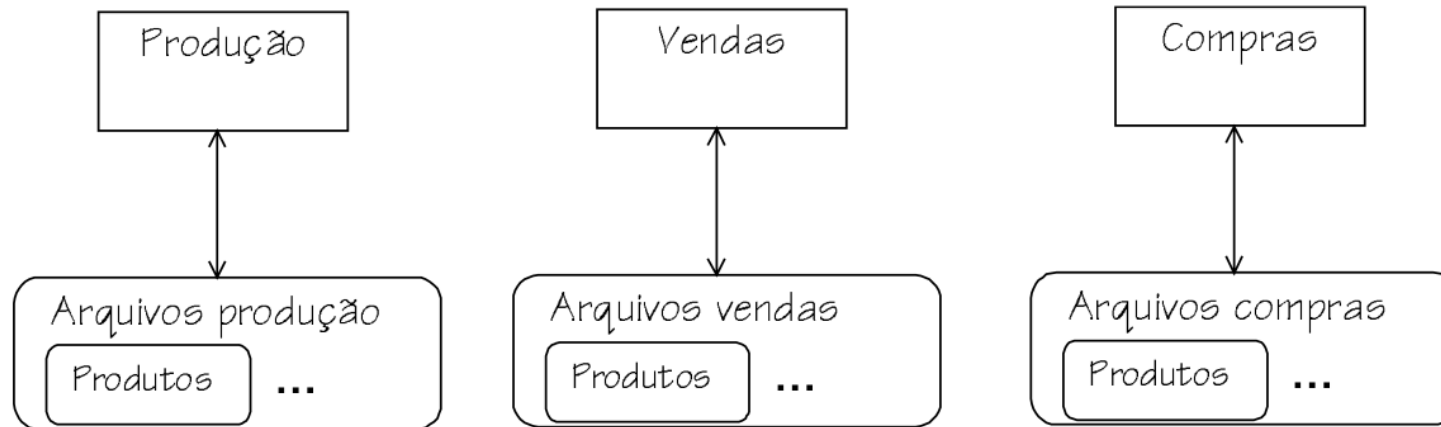
- Considere um sistema de uma indústria onde três setores (Produção, Vendas e Compras) precisam dos dados dos produtos que são produzidos nela para realizar suas funções.
 - A imagem abaixo ilustra uma situação onde em que cada setor possui sua própria cópias dos dados dos produtos em seus arquivos.



- Se cada uma das funções da indústria for informatizada de forma separada, sem considerar as demais funções, pode ocorrer a criação de arquivos repetidos com os dados dos produtos.
 - Nesse caso, surge o problema da **redundância de dados**:
 - A redundância de dados ocorre quando uma mesma informação está representada várias vezes em um sistema.

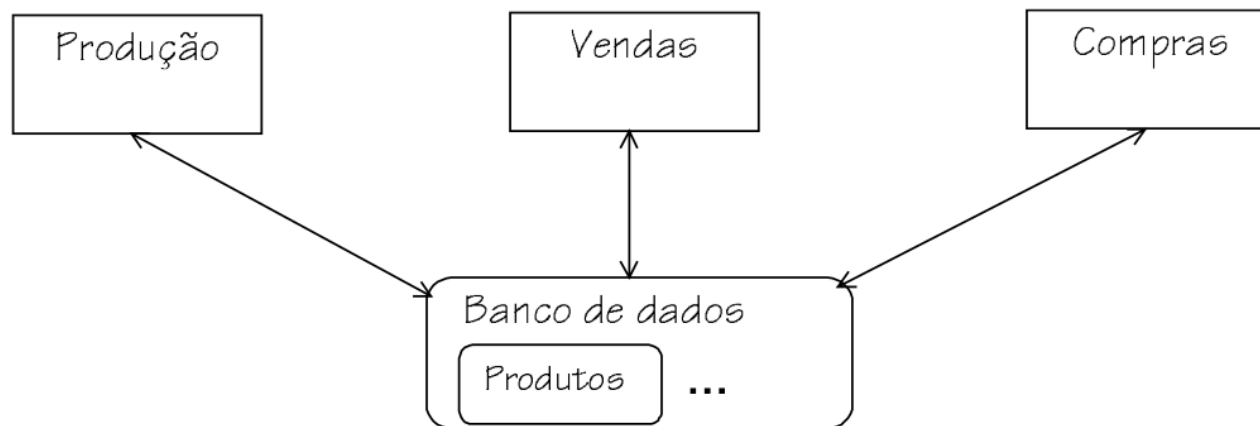
Controle de redundância

Como o problema da redundância poderia ser evitado na indústria?



Controle de redundância

Como o problema da redundância poderia ser evitado na indústria?



- Utilizando uma mesma base de dados no processo de informatização de seus processos.
 - Assim, cada informação é armazenada uma única vez, evitando da múltiplas cópias dos dados dos produtos.
 - Quando algum dos vários sistemas dessa indústria precisar acessar uma informação, ele fará acesso ao banco de dados que contém todas as informações.

Controle de redundância

Tipos de redundância

- **Redundância controlada:**

- O software tem conhecimento da múltipla representação da informação e garante a sincronia entre as suas diversas representações.
- Esse tipo de redundância é utilizada para melhorar a confiabilidade ou o desempenho global do sistema.
- **Exemplo:** Em um sistema bancário, é possível saber o saldo de um cliente a partir do histórico de todas as transações realizadas. Porém, podemos manter o seu saldo armazenado, atualizando-o sempre que uma transação é realizada. Assim, a consulta do saldo será realizada de forma muito mais rápida.

- **Redundância não controlada:**

- A responsabilidade pela manutenção da sincronia entre as diversas representações de uma informação está com o usuário e não com o software.
- Esse tipo de redundância deve ser evitada.

Controle de redundância

Tipos de redundância

Por que evitar a redundância não controlada?

- **Entrada repetida de uma mesma informação:**
 - A entrada repetida de uma mesma informação exige trabalho desnecessário. Além disso, pode resultar em erros de transcrição de dados.
- **Inconsistência de dados:**
 - Como a responsabilidade de manter a sincronia é do usuário, pode ocorrer erros na operação (alterar um dado sem alterar suas cópias ou demais representações).

Evolução dos SGBDs

Um breve histórico...

- **1960** - Surgimento dos primeiros SGBDs (CODASYL e IMS).
- **1970** - Modelo Relacional é proposto e publicado por Edgar Frank Codd.
- **1974** - Surgem os primeiros produtos (Igres da UBD e System R da IBM).
- **1976** - Modelo Entidade Relacionamento é proposto por Peter Chen.
- **1980** - O modelo relacional se torna popular e a linguagem SQL se torna um padrão.
- **1990** - Surgem os bancos de dados orientados a objetos e objeto-relacionais.
- **1990** - Popularização dos armazéns de dados (*data warehouse*).
- **2000** - Surge o NoSQL como proposta para problemas diferentes de representação de dados.
- **2010** - NewSQL (características dos sistemas relacionais e NoSQL).

Dúvidas?

André L. Maravilha

andre.maravilha@cefetmg.br
<https://andremaravilha.github.io/>

