

## **MODELAGEM DE DADOS**

**PROF. RAFAEL DIAS RIBEIRO, M.Sc.**  
**@ribeirord**

## MODELAGEM DE DADOS

Aula 2

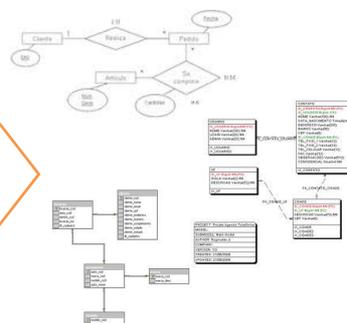
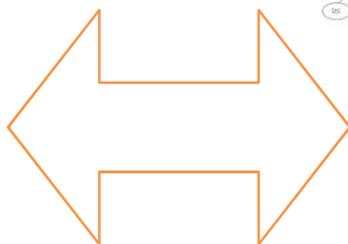
Prof. Rafael Dias Ribeiro. M.Sc.  
@ribeirord

### Objetivos:

- Revisão sobre Banco de Dados e SGBDs
- Aprender as principais características dos SGBDs.
- Aprender quando empregar e quando não empregar bancos de dados
- Conhecer conceitos fundamentais de um ambiente com SGBD

### Bancos de Dados

De acordo com (Navathe, 2005), podemos definir um banco de dados como **um conjunto de dados que se relacionam**.



### **Bancos de Dados**

Porém, o significado do termo é mais restrito do que esta definição. Um banco de dados, necessariamente, possui as seguintes propriedades:

- um banco de dados é uma coleção lógica coerente de dados com um significado inerente;
- uma disposição desordenada de dados não pode ser referenciada como um banco de dados;
- um banco de dados é projetado, construído e populado com dados para um propósito específico;

### **Bancos de Dados**

Porém, o significado do termo é mais restrito do que esta definição. Um banco de dados, necessariamente, possui as seguintes propriedades:

- um banco de dados possui um conjunto pré definido de usuários e aplicações;
- um banco de dados representa algum aspecto do mundo real, o qual é chamado de “mini-mundo” e qualquer alteração efetuada neste mini-mundo é automaticamente refletida no banco de dados.

### **Sistemas Gerenciadores de Bancos de Dados e Sistemas de Banco de Dados**

- Um SGBD é uma coleção de programas que permite aos seus usuários criarem e manipularem bancos de dados.
- Um SGBD é conjunto complexo de software que deve prover um conjunto básico de funcionalidades.

Na abordagem utilizando um SGBD, um único Banco de Dados é criado, mantido e acessado por todos os sistemas. Algumas características do SGBD:

- Natureza auto descritiva
- Isolamento entre os programas e os dados, e a abstração de dados
- Suporte para as múltiplas visões de dados
- Compartilhamento de dados e processamento de transações de multiusuários

### **Natureza auto descritiva**

- Descrição completa da estrutura do banco e suas restrições
  - Estrutura de cada item, tipo e formato de cada dato, restrições
- Existe um catálogo que armazena as informações chamado metadados
  - O catálogo é utilizado pelo próprio SGBD e também pelos usuários que precisam de informações sobre a estrutura do banco.
  - Não se altera as informações de metadados, o uso de tais informações é apenas para consulta, caso contrário, o banco poderá apresentar graves problemas.
- O SGBD precisa trabalhar com uma diversidade muito grande de aplicações, o catálogo permite isso.

### **Isolamento entre os programas e os dados, e a abstração de dados**

- Diferente do processamento de arquivos, os programas não possuem informações sobre o armazenamento dos dados
- A estrutura dos arquivos de dados é armazenada nos catálogos do SGBD
- O Banco de Dados pode ter sua estrutura alterada sem precisar alterar os programas que o acessam
- Essa característica é chamada **independência programa-dados**

### **Isolamento entre os programas e os dados, e a abstração de dados**

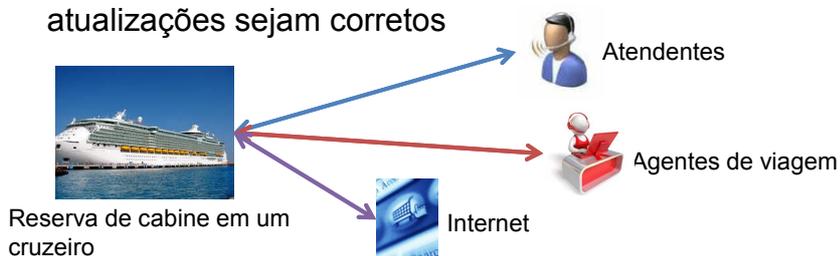
- O SGBD fornece aos programas uma representação conceitual de dados, que não inclui detalhes de armazenamento
- Essa característica é chamada de **abstração de dados**

### **Suporte para as múltiplas visões de dados**

- Um banco de dados típico tem muitos usuários e cada usuário pode precisar de diferentes visões do banco de dados.
- Uma visão pode ser:
  - Um subconjunto do banco de dados, que atendem critérios de necessidades do usuário
  - Uma visão virtual dos dados, derivada dos dados existentes, simulando dados/informações que não estão explicitamente armazenados

## Compartilhamento de dados

- Um SGBD deve permitir o acesso concorrente de diversos usuários
- O controle de concorrência deve permitir que muitos usuários, ao tentar atualizar o mesmo dado, o façam de modo controlado, para assegurar que os resultados das atualizações sejam corretos



## Processamento de transações

- **Transação:** é um processo que inclui um ou mais acessos no banco de dados, como leitura e gravação de registros
  - **Isolamento:** garante que cada transação possa ser efetuada de forma isolada de outras transações
  - **Atomicidade:** garante que todas as operações da transação sejam realizadas ou nenhum delas seja
    - Exemplo: Transferência bancária
      - Leitura de saldo
      - Validação de saldo
      - Débito da conta origem
      - Validação da conta de destino
      - Crédito da conta destino
- O SGBD deve permitir que transações concorrentes operem corretamente

### **Independência de Dados**

- “Consiste na capacidade de permitir que haja evolução na descrição dos dados da empresa, sem que os sistemas ou aplicações tenham que ser alterados”
- “Representa a imunidade das aplicações às mudanças na estrutura de armazenagem e estratégias de acesso”

### **Independência de Dados**

- A independência dos dados, em relação a aplicação, representa o um avanço no sentido de tornar qualquer modificação das estruturas dos arquivos imperceptível para os programas. Desta forma, as mudanças na organização dos dados não geram qualquer necessidade de alteração dos programas que o manipulam. Por exemplo:
  - Inclusão de um novo campo (atributo) em um arquivo (tabela) é feita sem que sejam alterados os programas, pois as estruturas dos arquivos não estão definidas nos programas.

### **Controle de Redundância**

- “Redundância é armazenar o mesmo dado várias vezes para atender diversas aplicações. Para manter a consistência do banco de dados, deve-se armazenar o dado uma única vez e em apenas um lugar no banco de dados. Isto permite manter a consistência, economizar espaço de armazenamento.” Em alguns casos, a redundância é necessária, porém ela deve ser controlada pelo sistema de gerenciamento de banco de dados. “ (Elmasri & Navathe, 2005)

### **Controle de Redundância**

- “É um conceito representado pelo controle centralizado dos dados compartilhados por diversas aplicações, reduzindo a repetição de dados a um mínimo justificável e aceita apenas por questão de desempenho.” (Cerícola, 1991)

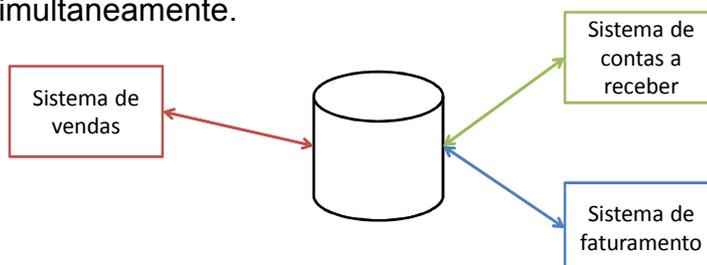
## Controle de Redundância

### Problemas da redundância de dados:

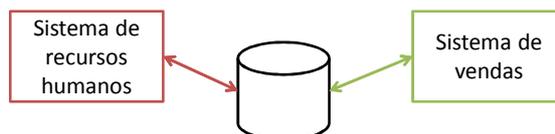
- Duplicação de esforço para manter os dados atualizados;
- Desperdício de espaço de armazenamento;
- Possibilidade de inconsistência dos dados

## Compartilhamento de Dados

- Permitir a usuários diferentes a utilização simultânea de um mesmo dado. Por exemplo:
  - As informações sobre clientes podem ser acessadas pelo sistema de vendas, de contas a receber e faturamento simultaneamente.



## Compartilhamento de Dados



- A mesma base de dados sobre empregados pode ser usada simultaneamente pelo sistema de recursos humanos e pelo sistema de vendas.
- No primeiro caso os dados serão utilizados no processo de pagamento e no segundo no processo de alocação dos vendedores às áreas de atendimento a cliente.

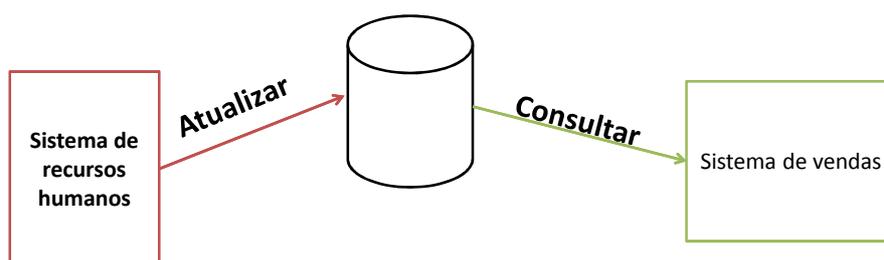
## Restrições de Acesso

“Um SGBD deve prover controles de segurança e autorização, que são utilizados para criar contas e seus respectivos direitos de acesso quando múltiplos usuários compartilham um banco de dados.” (Navathe,2005)

## Restrições de Acesso

“ Os controles de segurança abrangem conceitos tais como: procedimentos de validação e controle, garantia de integridade e controle de acesso, que visam resguardar o banco de dados de uma possível perda ou destruição de dados seja por falha de programa ou por falha de equipamento” (Cerícola, 1991)

## Restrições de Acesso



## Restrições de Acesso Não Autorizado

- Como todas os dados vão estar em um único local, é possível que nem todas as pessoas tenham acesso à todos os dados.
  - Uma pessoa da secretaria pode não ter acesso aos dados financeiro
  - Uma pessoa do financeiro pode não ter acesso aos dados de salários
- O SGBD deve ter um mecanismo de **segurança e autenticação**
  - Criação de usuários/grupos e autenticação por senhas
- Atribuição de uma série de **permissões/bloqueios**, mais comuns:
  - Apenas consultar
  - Consultar apenas alguns dados
  - Não apagar dados
  - Somente inserir dados
  - A granularidade das permissões depende do SGBD

## Restrições de Integridade

A maioria dos SGBDs proveem certas restrições de integridade que devem ser aplicadas aos dados. O SGBD deve ter mecanismos para possibilitar a definição das restrições e assegurar o respeito a estas. Um exemplo de restrição de integridade é a definição de um tipo de dado (data type) para cada item de dado.

### Restrições de Integridade

- O SGBD aceitará apenas os tipos definidos ou seja azul, vermelho, amarelo e verde como válidos para este atributo.

Cor
azul
vermelho
amarelo
verde

Peça	Cor

- Se o usuário tentasse informar, por exemplo, o valor branco para o atributo cor da parede, o sistema acusaria um erro.

### Mecanismos de Backup e Recuperação

- Um SGBD deve prover facilidades para recuperação de falhas do hardware ou software.
- Estes mecanismos evitam que cada aplicação tenha que projetar e desenvolver seus próprios controles contra a perda de dados.

## Mecanismos de Backup e Recuperação

- **Falha:** Ocorre no nível mais baixo do hardware ou software.
  - Uma fonte queimar
  - Uma linha de código errada
- **Erro:** Uma falha pode gerar um erro. Um erro é a representação da falha no universo da informação (dados).
  - Por consequência de uma falha um dado foi corrompido
- **Defeito:** O defeito é o que é percebido pelo usuário, é a representação de um erro no universo do usuário.
  - O sistema já está com erro e o dado já está corrompido

## Mecanismos de Backup e Recuperação

### Exemplo:

Se o sistema falha no meio de um programa de alteração complexo, o mecanismo de recuperação é responsável por assegurar que o banco de dados será restaurado para o estágio que ele se encontrava antes do início da execução do programa.

## **Múltiplas Interfaces**

- Um ambiente de banco de dados é acessado por variados tipos de usuários com variadas necessidades de informação e com diferentes níveis de conhecimento técnico.

## **Múltiplas Interfaces**

- Para atender esta diversidade usuários, o SGBD deva fornecer diferentes tipos de interfaces. Sendo assim este ambiente disponibiliza:
  - Linguagens de consulta para usuários casuais;
  - Linguagens de programação para programadores de aplicações;
  - Interfaces gráficas com formulários (telas) e menus para usuários paramétricos;
  - Interfaces para administração do banco de dados;
  - Linguagem natural.

### **Benefícios no uso de SGBDs**

- Os ambientes de bancos de dados fornecem uma série de vantagens na sua adoção:
  - Potencial para o estabelecimento e o cumprimento de padrões
    - Garantir o uso de padrões na organização, códigos, nomes, datas
  - Flexibilidade de mudanças
    - Alteração da estrutura dos dados de forma simples

### **Benefícios no uso de SGBDs**

- Redução no tempo de desenvolvimento de novas aplicações
  - Encapsulamento das regras de armazenamento e integrações
- Disponibilidade de informação atualizada
  - Assim que um dado é atualizado, ele está disponível para todos os usuários
- Economia de escala
  - Investimento pontual em profissionais e equipamentos

### **Bancos de dados NÃO são sempre a solução...**

- Apesar das vantagens de utilização, a escolha por um ambiente de banco de dados tem um alto custo atrelado.
- A sua adoção deve, então, compensar ou ser compatível com este custo.

### **Bancos de dados NÃO são sempre a solução...**

#### **Sobrecustos vinculados**

- Alto investimento inicial em software, pela aquisição do banco de dados e licenças, e em hardware que suporte este ambiente.
- Custo da generalidade do SGBD, ou seja, na definição e no processamento dos dados.
- “Overhead” de processamento.

### **Bancos de dados NÃO são sempre a solução...**

- Overhead significa tudo aquilo que o SGBD tem que fazer além de gerenciar os dados. Isto envolve tarefas tais como:
  - garantir segurança
  - controlar concorrência (utilização do mesmo dado por aplicações e usuários distintos simultaneamente)
  - recuperação de falhas
  - garantia de integridade.

### **Quando NÃO utilizar bancos de dados...**

- Volume de dados pequeno, aplicações simples, bem definidas.
- Mudanças não são esperadas.
- Ambientes de sistemas que exijam resposta em tempo real.
- Acessos múltiplos e concorrentes não são necessários.