

# Visão Geral de Gerenciamento de Banco de Dados

## Conceitos, Arquitetura e Modelagem

Prof. Elton Sarmanho<sup>1</sup>  
eltonss@ufpa.br

<sup>1</sup>Faculdade de Sistemas de Informação - UFPA Campus Cametá

16 de dezembro de 2025

## Roteiro da Aula

Conceitos Fundamentais

- Dados e Informação

- Conceitos Básicos

- Finalidades do SGBD

- Sistema de Gerenciamento de BD (SGBD)

Arquitetura e Modelagem

Linguagem SQL

Gerenciamento de Transações

Tópicos Avançados e Papéis

Conclusão e Referências

# Sistemas de Informação e Dados

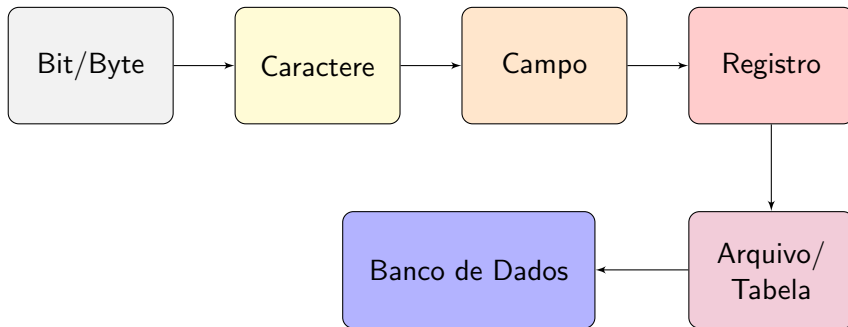
## O que é um Sistema de Informação?

Conjunto organizado de pessoas, hardware, software, redes e recursos de dados que coleta, transforma e dissemina informações em uma organização .

- ▶ **Dados:** Fatos brutos, observações ou medições (ex: temperatura, valor de venda).
- ▶ **Informação:** Dados processados e organizados de forma a terem significado e utilidade para o usuário .
- ▶ **Banco de Dados (BD):** Coleção de dados inter-relacionados, representando informações sobre um domínio específico.

## Hierarquia de Dados

Os dados são organizados logicamente do menor para o maior elemento :



## Caractere

- ▶ Elemento lógico mais simples dos dados: um **símbolo** (letra, dígito, pontuação).
- ▶ **Camada física**: caracteres são codificados (ex.: UTF-8) e ocupam **bytes**.
- ▶ **Camada lógica**: para o usuário, o caractere é a menor unidade manipulável em textos.

## Campo ou item de dados

- ▶ **Campo** (ou **atributo**) é um agrupamento de caracteres com significado.
- ▶ Exemplos: nome, email, data\_nascimento, total\_venda.
- ▶ Em modelagem de dados, um campo descreve uma característica de uma **entidade** (ex.: Cliente).

## Registro

- ▶ **Registro** (ou **tupla/linha**) é uma coleção de atributos que descrevem **uma ocorrência** de uma entidade.
- ▶ Exemplo: um registro de Funcionario com `matricula=123`, `nome='Ana'` e `salario=4500.00`.

## Arquivo

- ▶ Uma coleção de registros afins forma uma **tabela** (no modelo relacional) ou um **arquivo** (na visão de armazenamento).
- ▶ Exemplo: a tabela Funcionarios contém todos os registros de funcionários.
- ▶ Podem ser classificados quanto a sua permanência como:
  - ▶ **Arquivo mestre**: dados correntes/operacionais.
  - ▶ **Arquivo histórico**: dados antigos para auditoria/análise.



## Banco de Dados

- ▶ **Banco de Dados (BD)**: coleção organizada e relacionada de dados, descrita por um **esquema** e gerenciada por um **SGBD**.
- ▶ Um BD inclui: **tabelas**, **restrições**, **índices**, **visões** e **metadados**.
- ▶ Em geral, dados em um BD
  - ▶ podem ser **compartilhados** por múltiplas aplicações.
  - ▶ suportam **independência de dados** (em especial no nível físico).

**banco de dados**

	id [PK] integer	bairro character var	celular character var	endereco character var	enderecoprofil character var	estadocivil character var
1	5	Tapanã	-	Conj. Antonio G	eeeeeeeeeeee	Casado
2	4	Tapanã	2222-2222	conj. Aldo Almei	xxxxxxxxxxxxxx	Solteiro
3	5	Tapanã	2222-2222	Rua Almirante Tamar	xxxxxxxxxxxxxx	Solteiro
4	6	Tapanã			xxxxxxxxxxxxxx	Solteiro
5	7	Tapanã	1111-1111	Residencial Par. Rod. Mario Cov		Solteiro
6	8	Tapanã	1111-1111	Almirante Tamar	xxxxxxxxxxxxxx	Casado
7	9	Tapanã	8356-2556	Rua pte. Costa	xxxxxxxxxxxxxx	Casado
8	10	Tapanã	2222-2222	5ª Rua do Tape	96293126	Casado
9	11	Pratinha	2222-2222	Rua Felicidade, 4	xxxxxxxxxxxxxx	Solteiro
10	12	-----	2222-2222	PA 150 KM 50 -	xxxxxxxxxxxxxx	Solteiro
11	13	S. Vicente	8170-3511	Tr.Nazaré, 1023	pqueresma71@	Solteiro
12	14	Tapanã	-	Rua Paulo Guilh	ssssssssssssss	Solteiro
13	15	Bengui	8704-			
14	17	Tapanã	8105-			
15	18	Tapanã	8141-9884	Q-E 5 casa 15	xxxxxxxxxxxxxx	Casado
16	19	Tapanã	8111-2704	Rua São Clemen	SESMA	Solteiro
				"	xxxxxxxxxxxxxx	Solteiro
				Conj. Kikuche, A	xxxxxxxxxxxxxx	Solteiro
19	22	guama	3333-3333	sao domingos, 6	xxxxxxxxxxxxxx	Solteiro
20	23	Tapanã	8108-9055	Conj. Antonio G	eeeeeeeeeeee	Solteiro
21	25	Tapanã	-	Jardim Primavera	xxxxxxxxxxxxxx	Solteiro
22	26	Ponta Grossa-Id		Rua Coronel Ju	xxxx	Solteiro
23	27	Tapanã	8358-5025	Estrada do Tape	xxxxxxxxxxxx	Solteiro
24	28	Preatinha	2222-2222	Conj. Paulo Fon	eeeeeeeeeeee	Solteiro
25	29	Tapanã	8269-6534	Conj. Al. 9, 110	xxxxxxxxxxxxxx	Casado
26	30	Tapanã	8316-4684	Conj. Antonio G	xxxxxxxxxxxxxx	Solteiro
27	31	Tapanã	8313-1707	Rua Irmã Adelaide	ssssssssssssss	Solteiro

**colunas = campos**

**linhas = registros**

**tabelas de dados**

- ▶ Chave Primária (*primary key*)
  - ▶ Atributo(s) que **identifica(m) unicamente** cada registro de uma tabela.
  - ▶ Propriedades desejáveis: **unicidade**, **não nulo** e **estabilidade** (não mudar com frequência).
- ▶ Chave Estrangeira (*foreign key*)
  - ▶ Atributo(s) em uma tabela que **referencia(m)** a PK (ou chave candidata) de outra tabela.
  - ▶ É o mecanismo central para modelar relacionamentos e impor **integridade referencial**.

## Reflexão

Em um sistema de pedidos, por que `pedido.cliente_id` deve ser uma FK para `cliente.id`? O que pode dar errado sem isso?

## Integridade Referencial

- ▶ Garante consistência entre tabelas relacionadas via **chaves estrangeiras**.
  - ▶ Exemplo 1: o banco de dados não permite cadastrar um pedido para um cliente que ainda não existe.
  - ▶ Exemplo 2: não é permitida a exclusão de clientes que possuem pedidos cadastrados (ou define-se CASCADE), **evitando registros órfãos**.

## Integridade Referencial (visual)

Cliente

codigoCliente	Nome	endereço	codigoCidade
548	Maria	Rua Carvalho 615	1
549	Pedro	Rua Pedro Chaves 22	5

Viola a restrição → cidade 5 não existe

Cidade

codigoCidade	Descricao	Estado
1	Florianópolis	SC
2	São José	SC

## Integridade Referencial (exemplo em SQL)

```
1 CREATE TABLE cidade (  
2     id      INT PRIMARY KEY,  
3     nome   VARCHAR(80) NOT NULL  
4 );  
5  
6 CREATE TABLE cliente (  
7     id          INT PRIMARY KEY,  
8     nome        VARCHAR(120) NOT NULL,  
9     cidade_id   INT NOT NULL,  
10    CONSTRAINT fk_cliente_cidade  
11        FOREIGN KEY (cidade_id)  
12        REFERENCES cidade(id)  
13        ON UPDATE CASCADE  
14        ON DELETE RESTRICT  
15 );
```

## Domínio (*domain*)

- ▶ **Domínio** é o conjunto de valores permitidos para um atributo (tipo + regras).
- ▶ Exemplo: o atributo sexo pode ter domínio { 'M', 'F' } (ou outro conjunto definido pelo requisito).
- ▶ Em SQL, domínios aparecem como **tipos** e **constraints** (NOT NULL, CHECK, etc.).

## Stored Procedures

- ▶ São funções que estão armazenadas no banco de dados e que são **executadas diretamente** no servidor de banco de dados.



## Triggers

- ▶ É um recurso de programação presente na maioria dos SGBD, utilizado para **associar** um **procedimento armazenado** a um **evento** do banco de dados (inclusão, alteração e exclusão) de modo que o procedimento armazenado seja executado **automaticamente** sempre que o evento associado acontecer.

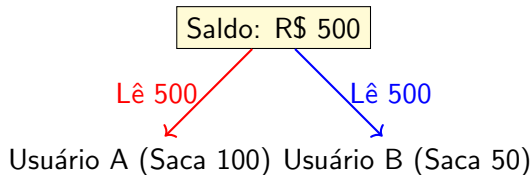
- ▶ Suprimir problemas de um **sistema de processamento de arquivos**
  - ▶ Redundância: vários arquivos com as mesmas informações
    - ▶ Não há um mecanismo para estabelecer o relacionamento
  - ▶ Inconsistência: dados que são alterados em um arquivo e não em outros
  - ▶ Isolamento dos Dados: os dados estão dispersos em vários arquivos, e estes arquivos podem apresentar diferentes formatos
    - ▶ Provocando dificuldade em desenvolver novas aplicações

- ▶ Problemas de Integridade: programas devem garantir a manutenção de restrições de integridade (consistência).
  - ▶ Restrições de Integridade → regras que estabelecem quando uma base de dados está com dados (domínio) e relacionamentos (referencial) válida.
  - ▶ Não ocorrendo em sistemas de processamento de arquivos

- ▶ Problemas de atomicidade: um sistema de processamento está sujeito a falhas. Mecanismos simples como backups não são suficientes. Após falhas o banco de dados deve ser recuperado rapidamente, em seu **último estado consistente**.
  - ▶ Operação Atômica: deve ocorrer **por completo, senão cancela transação**.
  - ▶ Por exemplo, uma pessoa deve transferir R\$ 1000,00 da conta BanPará para uma conta Santander. Se ocorrer falha no sistema durante a transação, é possível que os R\$ 1000,00 sejam debitados da conta BanPará e não apareça conta Santander, criando um estado inconsistente no banco de dados

## Controle de Concorrência

Problema clássico: Dois clientes acessam a mesma conta ao mesmo tempo.



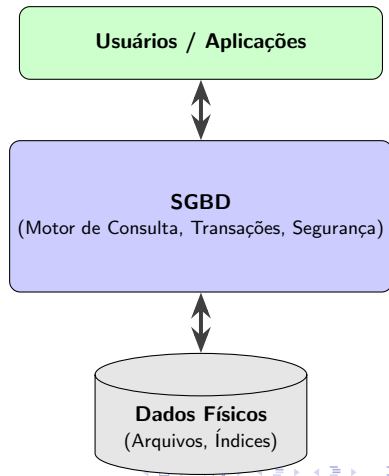
Sem controle de concorrência (Locking), o saldo final pode ficar inconsistente (R\$ 450 ou R\$ 400 ao invés de R\$ 350)!

- ▶ Problemas de Segurança: nem todos os usuários de banco de dados estão autorizados ao acesso a todos os dados.
  - ▶ Deve-se ter restrições a nível de usuário

A principal finalidade de um “SGBD” é proporcionar um ambiente, **conveniente e eficiente**, para a recuperação e armazenamento das informações do banco de dados.

## O que é um SGBD?

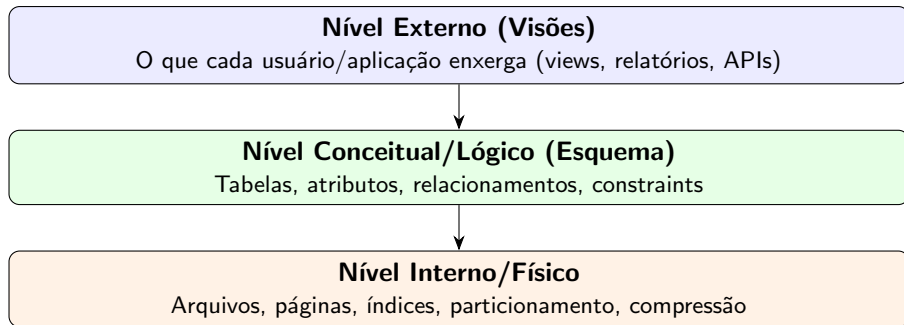
- ▶ **SGBD (DBMS):** Software de sistema para criar, gerenciar e usar bancos de dados.
- ▶ Atua como uma interface entre os **programas aplicativos** e os **arquivos de dados** físicos .
- ▶ Garante que os dados sejam independentes dos programas (Abstração de Dados) .



- ▶ A abordagem do gerenciamento de BD abrange três atividades fundamentais:
  - ▶ Atualização e manutenção de BD comuns para refletir novas transações de negócios e outros eventos que exigem mudanças nos registros de uma organização.
  - ▶ Fornecimento das informações requisitadas por cada aplicação do usuário final mediante o uso de aplicativos que compartilham os dados em BD comuns
  - ▶ Fornecimento de uma capacidade de consulta/resposta e emissão de relatório por meio de um pacote SGBD.

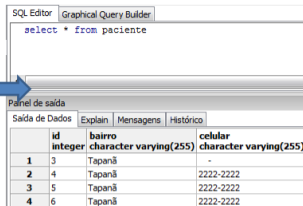


## Níveis de abstração



## Níveis de abstração

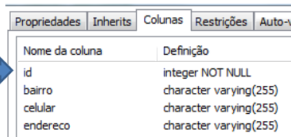
**Nível de visão (externo)**  
visão de cada usuário



The screenshot shows a database management system interface. At the top, there are two tabs: 'SQL Editor' and 'Graphical Query Builder'. The 'SQL Editor' tab is active, displaying the query `select * from paciente`. Below the query editor is a 'Panel de saída' (Output Panel) with four sub-tabs: 'Saída de Dados', 'Explain', 'Mensagens', and 'Histórico'. The 'Saída de Dados' tab is selected, showing a table of query results. The table has four columns: 'id' (integer), 'bairro' (character varying(255)), 'celular' (character varying(255)), and an unlabeled column. The results are as follows:

	id integer	bairro character varying(255)	celular character varying(255)	
1	3	Tapanã	-	
2	4	Tapanã	2222-2222	
3	5	Tapanã	2222-2222	
4	6	Tapanã	2222-2222	

**Nível lógico (conceitual)**  
quais dados estão  
armazenados no BD



The screenshot shows a database management system interface with a tabbed view. The tabs are 'Propriedades', 'Inherits', 'Colunas', 'Restrições', and 'Auto-v'. The 'Colunas' tab is selected, showing a table of column definitions. The table has two columns: 'Nome da coluna' (Column Name) and 'Definição' (Definition). The definitions are as follows:

Nome da coluna	Definição
id	integer NOT NULL
bairro	character varying(255)
celular	character varying(255)
endereco	character varying(255)

**Nível físico (interno)**  
como os dados estão de fato armazenados, por exemplo, bloco de "bytes"  
consecutivo de memória

## Por que usar um SGBD? (Problemas de Arquivos)

Sistemas de arquivos tradicionais sofrem de:

1. **Redundância:** Mesmos dados repetidos em vários arquivos .
2. **Inconsistência:** Alterar em um lugar e esquecer do outro.
3. **Isolamento:** Dificuldade em cruzar dados de arquivos diferentes.
4. **Integridade:** Difícil aplicar regras de negócio (ex: Saldo não pode ser negativo) .
5. **Atomicidade:** Falhas no meio de uma operação deixam dados corrompidos .

### Solução SGBD

O SGBD centraliza o controle, reduz redundância e garante integridade e segurança .

## Níveis de Abstração (Esquema vs. Instância)

- ▶ **Esquema (Schema):** A estrutura lógica do banco (tabelas, campos, tipos). Muda pouco .
- ▶ **Instância:** O conjunto de dados armazenados em um momento específico. Muda constantemente com INSERTs e DELETEs .

### Independência de Dados:

- ▶ *Física:* Alterar onde o dado está salvo (HD/SSD) sem quebrar o esquema lógico.
- ▶ *Lógica:* Adicionar campos numa tabela sem quebrar as aplicações existentes .

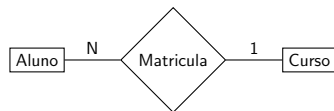
## Modelagem de Dados: As 3 Etapas

O projeto de um BD passa por três níveis de abstração :

1. **Modelo Conceitual:** Alto nível, focado no negócio. O que guardar?
  - ▶ Ferramenta: Diagrama Entidade-Relacionamento (DER) .
2. **Modelo Lógico:** Tradução para um tipo de BD (ex: Relacional).
  - ▶ Estrutura: Tabelas, Chaves Primárias (PK), Chaves Estrangeiras (FK) .
3. **Modelo Físico:** Detalhes de implementação.
  - ▶ Índices, tipos de dados específicos (VARCHAR, INT), particionamento .

## Exemplo: Conceitual → Lógico

### Conceitual (Entidade-Relacionamento)



### Lógico (Tabelas Relacionais)

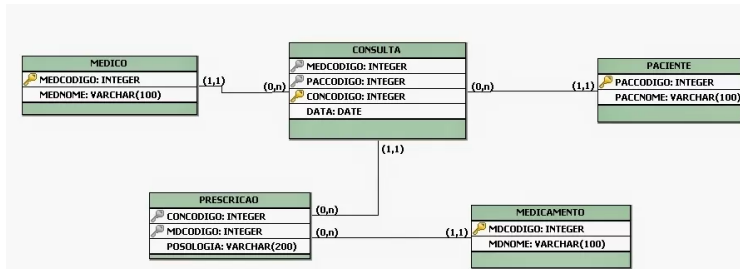
TB_ALUNO		
ID_ALUNO (PK)	Nome	ID_CURSO (FK)
101	João	1
102	Maria	1

TB_CURSO	
ID_CURSO (PK)	NomeCurso
1	Sistemas de Inf.

**Nota:** A **Chave Estrangeira (FK)** cria o elo entre as tabelas, garantindo a Integridade Referencial .

## Modelo Físico

- ▶ **Objetivo:** Descrever como os dados serão armazenados fisicamente no sistema de banco de dados, incluindo detalhes como índices, partições e estratégias de otimização.
- ▶ **Ferramentas:** Especificações detalhadas de esquema, considerações de desempenho e otimização.
- ▶ **Exemplo:** No contexto de um sistema de biblioteca, você pode especificar detalhes como o tipo de índices a serem usados, a alocação de espaço em disco e estratégias de particionamento para otimizar o desempenho.





# SQL: Structured Query Language

Linguagem padrão dividida em subconjuntos:

- ▶ **DDL (Data Definition Language):** Define a estrutura (Esquema).
  - ▶ Comandos: CREATE, ALTER, DROP .
- ▶ **DML (Data Manipulation Language):** Manipula os dados (Instância).
  - ▶ Comandos: SELECT, INSERT, UPDATE, DELETE .

## Exemplo Prático DDL e DML

### DDL - Criando a Estrutura

```
1 CREATE TABLE Clientes (  
2     ID INT PRIMARY KEY,  
3     Nome VARCHAR(100),  
4     Email VARCHAR(100)  
5 );
```

### DML - Manipulando Dados

```
1 -- Inserindo  
2 INSERT INTO Clientes VALUES (1, '  
    Elton', 'elton@ufpa.br');  
3  
4 -- Consultando  
5 SELECT Nome FROM Clientes WHERE ID  
    = 1;
```

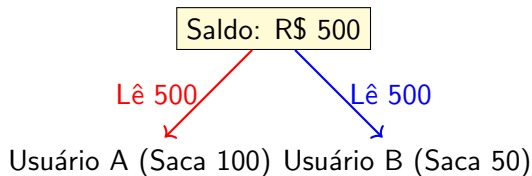
## Transações e Propriedades ACID

Uma **Transação** é uma unidade lógica de trabalho (ex: Transferência bancária). Para ser confiável, o SGBD garante 4 propriedades (ACID):

- ▶ **Atomicidade:** Tudo ou nada. Se falhar no meio, desfaz tudo (*Rollback*) .
- ▶ **Consistência:** O banco sai de um estado válido para outro válido .
- ▶ **Isolamento:** Transações simultâneas não interferem umas nas outras (Concorrência) .
- ▶ **Durabilidade:** Após o *Commit*, os dados estão salvos permanentemente, mesmo se faltar luz.

## Controle de Concorrência

Problema clássico: Dois clientes acessam a mesma conta ao mesmo tempo.



Sem controle de concorrência (Locking), o saldo final pode ficar inconsistente (R\$ 450 ou R\$ 400 ao invés de R\$ 350)!

## Papéis no Ambiente de BD

**DBA (Administrador):** O "dono" do BD. Define o esquema, segurança, acesso, backup e otimização .

**Desenvolvedor de Aplicação:** Cria programas (Java, Python, Mobile) que consomem o BD via DML .

**Usuário Final:** Usa o sistema (App Mobile, Site) sem saber que existe um BD por trás .

**Analista de Dados/BI:** Usa ferramentas de OLAP e Data Warehouse para tirar insights estratégicos .

## Além do Operacional: Data Warehouse e BI





Enquanto o BD relacional (OLTP) foca no dia a dia (vendas, cadastros), temos estruturas para análise:

- ▶ **Data Warehouse (DW):** Armazém de dados históricos integrados .
- ▶ **ETL (Extract, Transform, Load):** Processo de retirar dados do BD operacional, limpar e jogar no DW .
- ▶ **BI (Business Intelligence):** Painéis e relatórios para tomada de decisão gerencial .

## Resumo

1. SGBDs resolvem problemas de redundância e inconsistência de arquivos.
2. Dados são modelados em níveis (Conceitual  $\rightarrow$  Lógico  $\rightarrow$  Físico).
3. SQL é a linguagem universal (DDL para estrutura, DML para dados).
4. Transações (ACID) garantem a confiabilidade dos dados.

## Referências Bibliográficas

-  HEUSER, C. A. *Projeto de Banco de Dados*. 6<sup>a</sup> ed. Bookman, 2009.
-  SILBERSCHATZ, A. et al. *Sistema de Banco de Dados*. Campus.
-  DATE, C. J. *Introdução a Sistemas de Banco de Dados*. Campus.
-  ELMASRI, R.; NAVATHE, S. *Sistemas de Banco de Dados*. Pearson.