

FGA0137

Sistemas de Banco de Dados 1

Prof. Maurício Serrano

Material original: Profa. Elaine Parros Machado de Sousa
Prof. Jose Fernando Rodrigues Junior

2021/2

Introdução a Sistemas de Bancos de Dados

Módulo 1

Motivação

- Lawrence Joseph Ellison – **Oracle**
 - Homem mais rico do mundo (2000)
 - 9º americano mais rico (2022)
 - 10º pessoa mais rica do mundo (2022)
 - Bloomberg Billionaires Index e Forbes
 - \$108 bi em 2022
 - \$57 bi em 2018



Agenda

- Cronologia
- Evolução dos Sistemas de Informação
- Conceitos Básicos
- SGBDs

Cronologia: Década de 50

- Primeiros computadores.
- Programação **em linguagem de máquina**
- Surgimento de **SOs e Linguagens**
- Surgimento de **Estruturas de Dados**
- Sistemas de arquivos com **acesso aleatório**

1959: Sistema RAMAC (IBM)

Cronologia: Década de 60

- CODASYL: consórcio da indústria → **COBOL**
- Conceituação de **SGBD** e modelos de dados
- Primeiro SGBD comercialmente disponível
- **Modelo Hierárquico:**

IMS – IBM

Integração com a linguagem **COBOL**

Ainda **em uso corrente**

Cronologia: Década de 70

- Proposta do **Modelo Relacional**
- Surgimento de protótipos de **SGBD**
 - **INGRES** (UC – Berkeley)
 - **Sistema R** (IBM)
- Proposta do **MER**

Modelo Entidade-Relacionamento (MER)

Cronologia: Década de 80

- Primeiro SGBDR de grande porte disponível

DB2 – IBM:



- Surge SQL vinculada ao **Sistema R da IBM**
- SQL torna-se padrão
- Primeiros **Modelos Orientados a Objetos**

Cronologia: Década de 90+

- SGBDs orientados a objetos
 - O2
 - ObjectStore
 - Objectivity/DB
 - Jasmine
- SGBDs objeto-relacionais
 - Oracle8, Oracle9, Oracle10g, Oracle 11g
 - PostgreSQL
 - Informix

Cronologia: Década de 90+

- NOSQL orientados a documentos
 - MongoDB
 - CouchDB
 - Amazon SimpleDB
 - Lotus Notes
- NOSQL em grafos
 - Neo4J
 - Infinite Graph
 - FlockDB

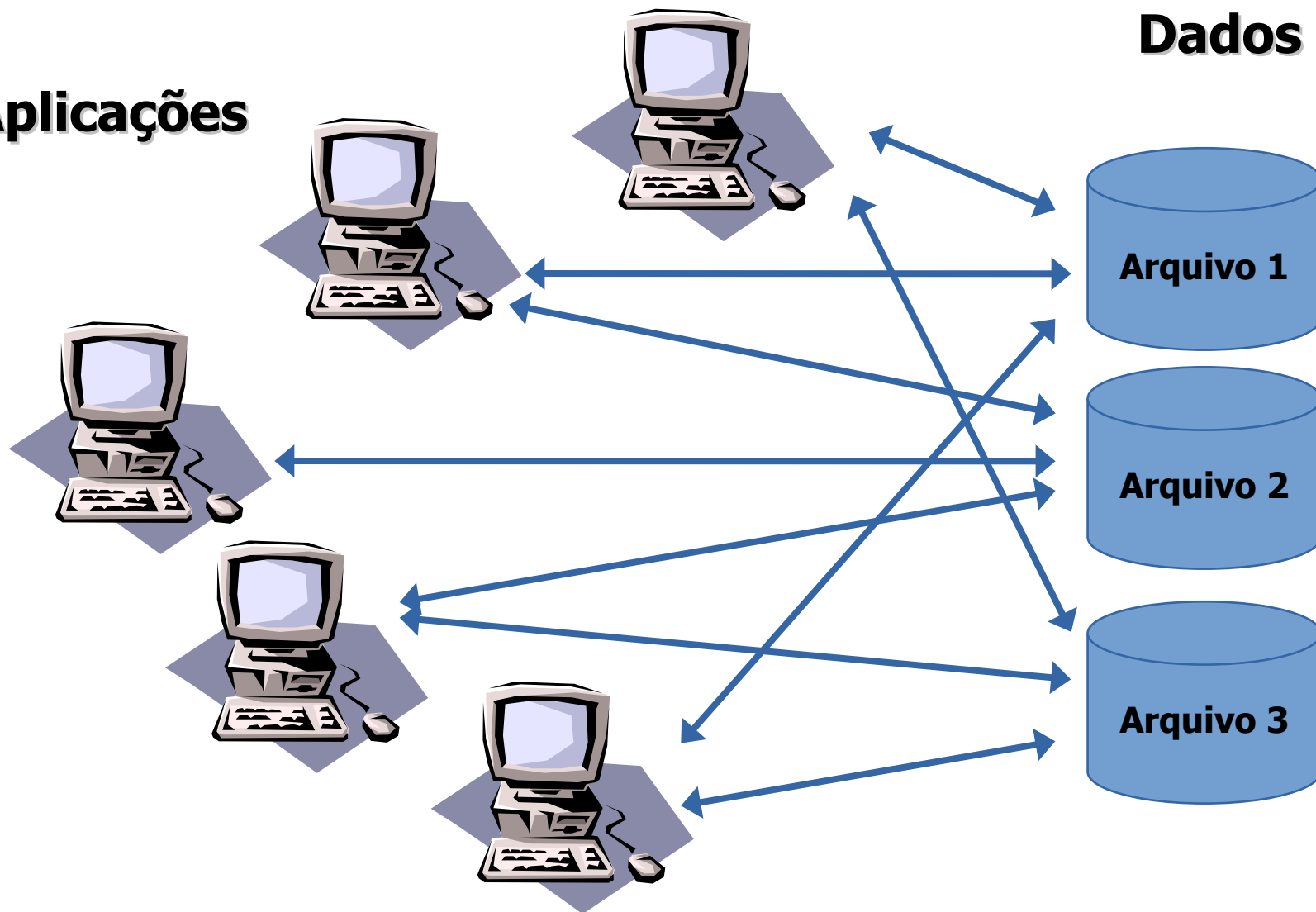
Evolução dos Sistemas de Informação

Sistemas de Informação baseados em gerenciamento de arquivos:

- Cada unidade da organização possui seus programas e arquivos
- Programas curtos para tarefas específicas
- Dados armazenados em disco
- Cada arquivo usa uma estrutura de dados

Aplicações

Dados

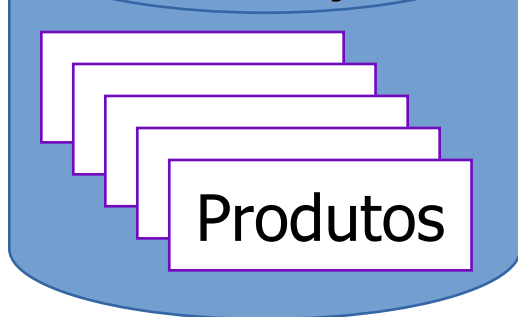


REDUNDÂNCIA ↔ INCONSISTÊNCIA

**Aplicação de
Produção**



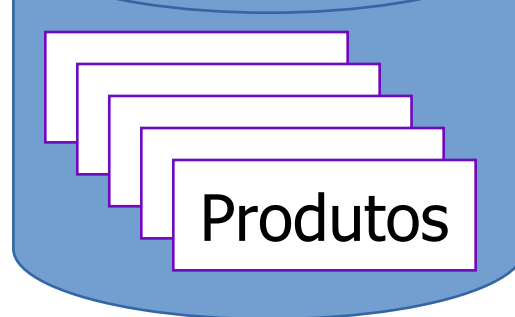
**Arquivos de Dados
de Produção**



**Aplicação de
Vendas**



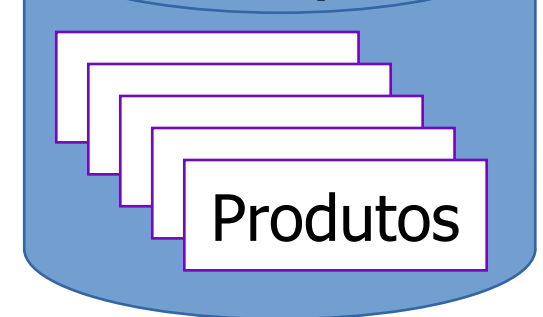
**Arquivos de Dados
de Vendas**



**Aplicação de
Compras**



**Arquivos de Dados
de Compras**



Consistência de Dados

- **Consistência** é o “estado ou caráter do que é coerente, do que tem solidez, veracidade, credibilidade, estabilidade, **realidade**”.
- Se determinada informação é replicada (redundância), seu valor é sempre o mesmo.

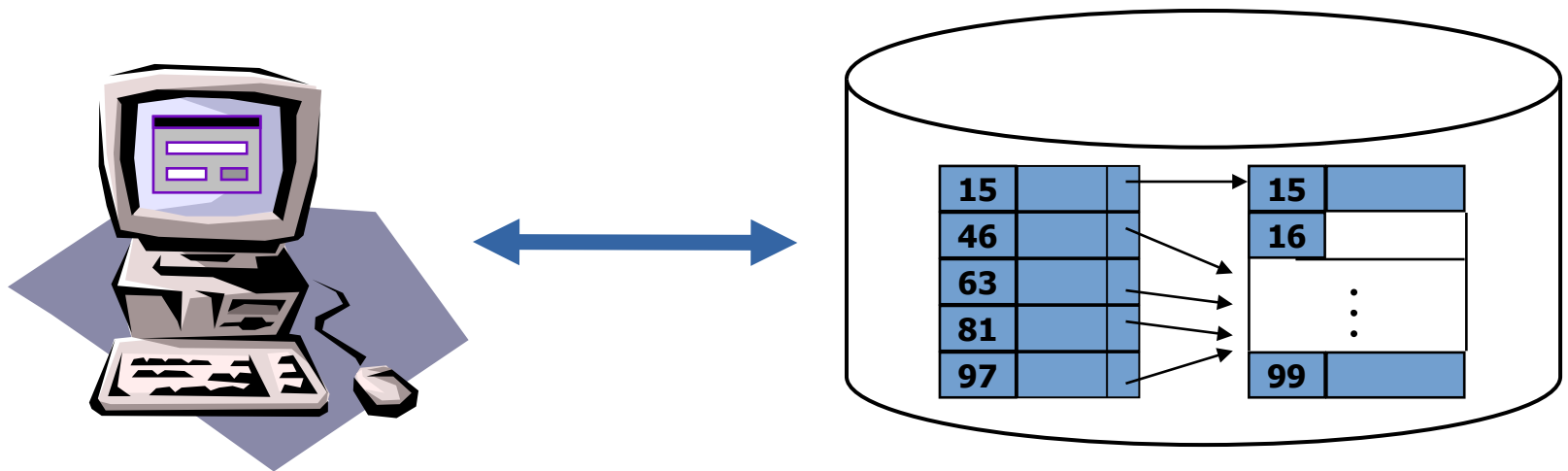
SIs baseados em arquivos

Problemas?

- Redundância e inconsistência de dados
- Dificuldade de acesso aos dados
- Isolamento de dados
- Anomalias no acesso concorrente
- Segurança

SIs baseados em arquivos

- Dados gravados em disco usando **ESTRUTURAS DE DADOS**
- Acesso requer conhecimento destas estruturas \Rightarrow **DEPENDÊNCIA DE DADOS.**



Dependência dos Dados

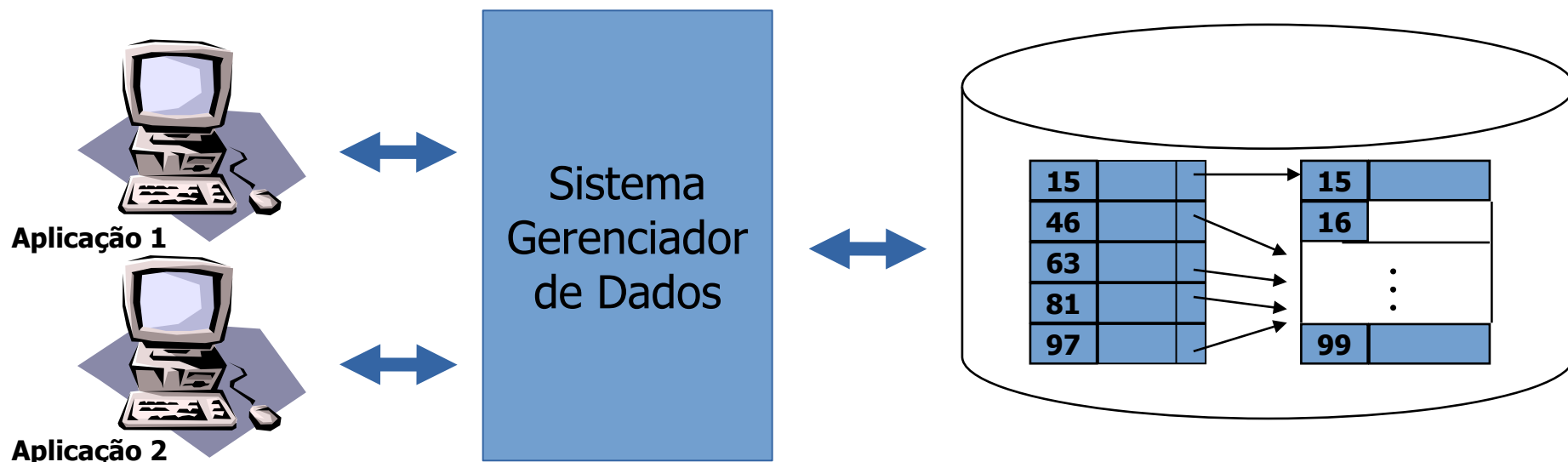
- Vários programas compartilhando os mesmos dados
- Todos devem conhecer e manipular as mesmas estruturas
- **ACOPLAMENTO FORTE**
- E se houver uma alteração na estrutura de dados?

TODOS OS PROGRAMAS TERÃO QUE SER ALTERADOS

Independência dos Dados

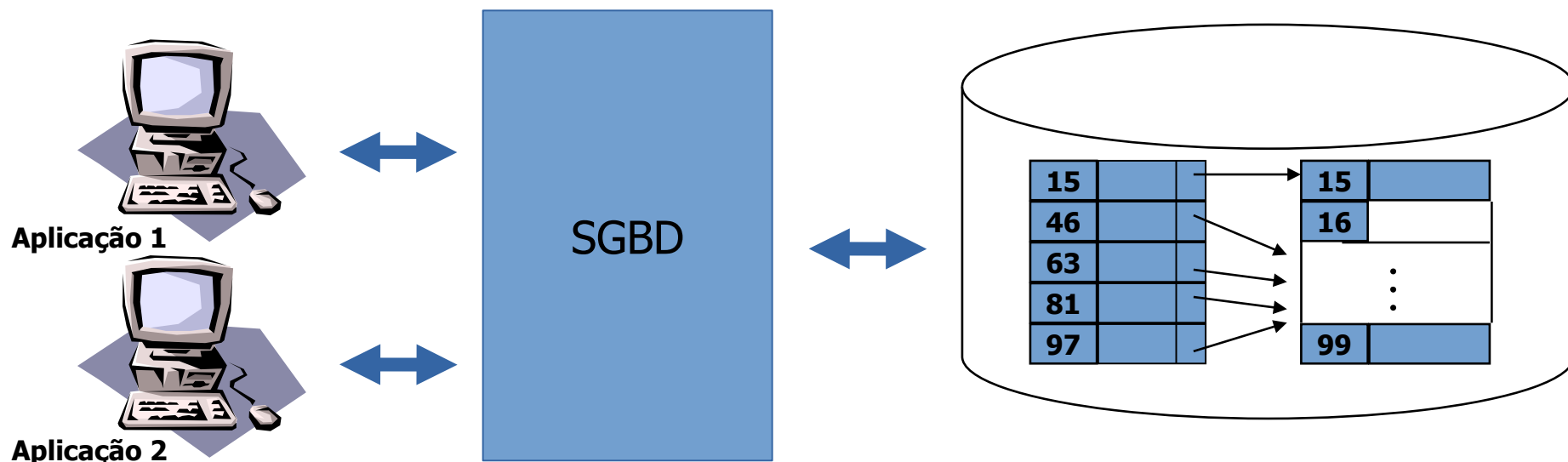
- Como tornar os programas **INDEPENDENTES** da estrutura de dados?

CRIANDO UM SISTEMA QUE GERENCIE A ESTRUTURA



Independência dos Dados

SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD)



SGBD

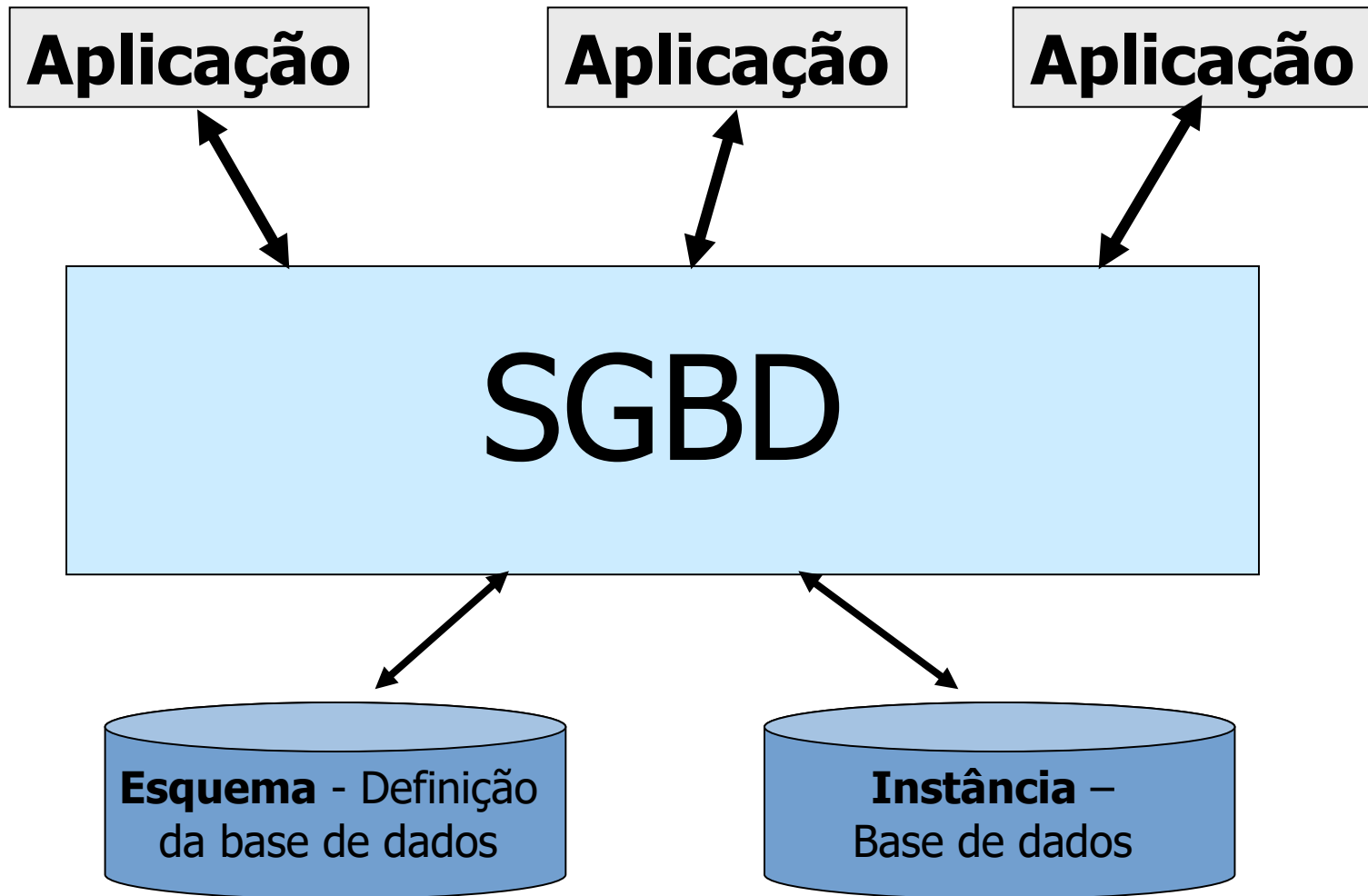
Composto por:

- Conjunto de Dados
 - Bases de Dados, Tabelas e Índices, Tuplas
- Conjunto de programas
 - Acesso e manipulação dos dados

SGBD

É um sistema de propósito geral:

- Mantém um conjunto lógico e organizado de dados
- Armazena grandes volumes de dados
- Permite busca e atualização dos dados
- É eficiente
- É autônomo em relação às aplicações



SGBDs

Requisitos Fundamentais:

- Segurança

- Física (mais comum no passado)
- Lógica
 - *Username*s e *password*s
 - Perfis de usuário

- Integridade

- Consistência
- Validade

SGBDs

Restrições de integridade:

- Definem o que é válido e o que não é válido
- Exemplos:
 - Um funcionário não pode pertencer a mais do que um departamento
 - O preço de venda de um produto deverá ser superior ao seu custo
 - O código de cada produto deve ser único

SGBDs

Requisitos Fundamentais:

- Recuperação / Tolerância a falhas
 - Transações atômicas
 - Registros de *Log*
 - *Backup*
- Controle da concorrência

Componentes de um SGBD



Utilizando um SGBD

1) Modelagem:

- Modelo Entidade/Relacionamento
- Modelo Relacional

2) Definição:

- SQL DDL

3) Instanciação:

- SQL DDL/DML

4) Uso:

- SQL DML

Pragmatismo!

Utilizando um SGBD

- A interface dos bancos de dados é definida pela linguagem declarativa SQL (DDL + DML)
- **Procedural vs Declarativo**
 - **Procedural:** exige especificação de quais dados são necessários, e como obtê-los
 - requer uma sequência específica de operações a serem executadas
 - ex.: linguagens de programação como C e Pascal, e a linguagem de projeto de bancos de dados álgebra relacional
 - **Não-Procedural (Declarativo):** exige apenas especificação de quais dados são necessários, e não de como obtê-los

Utilizando um SGBD

SQL - Data Definition Language (DDL)

- Conjunto de comandos para definição do esquema da base de dados
- Principais elementos:
 - *create*
 - *alter*
 - *drop*
- Compilador/Interpretador DDL

Utilizando um SGBD

SQL - Data Manipulation Language (DML)

- Conjunto de comandos para manipulação dos dados de maneira compatível com o esquema
- Principais elementos:
 - *select*
 - *insert*
 - *delete*
 - *update*
- Compilador/Interpretador DML

Utilizando um SGBD

Metadados ou Dicionário de Dados:

- banco de dados do sistema
- armazena descrição do esquema
- armazena metadados
- armazena restrições de segurança e integridade
- outras denominações:
 - catálogo de dados
 - diretório de dados

