



# SCC-540 Bases de Dados

Profa. Elaine Parros Machado de Sousa

Prof. Moacir P Ponti Jr

## Introdução



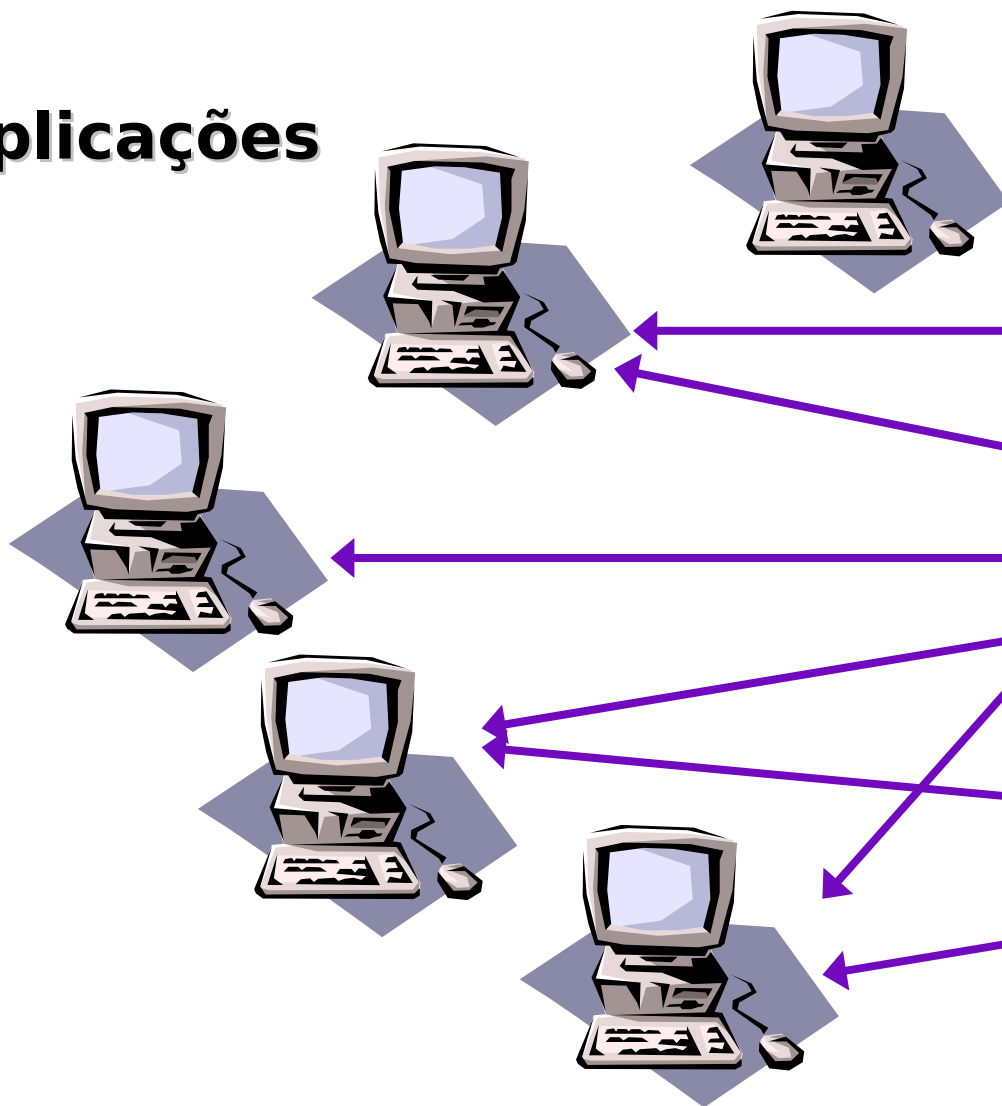
# Tópicos da aula

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

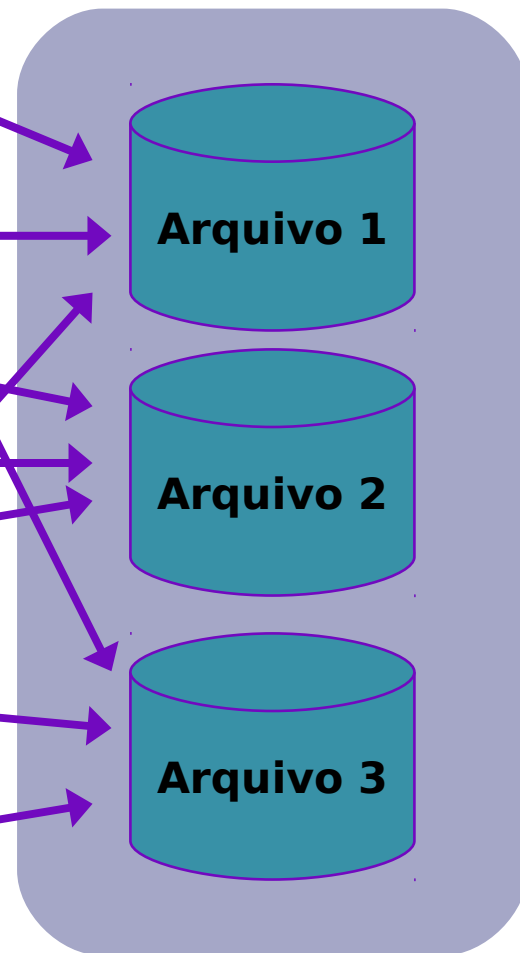
# Evolução dos Sistemas de Informação

- Sistemas de Informação baseados em gerenciamento de arquivos
  - programas e arquivos orientados a cada unidade organizacional
  - rotinas específicas para tarefas específicas
  - dados armazenados em disco, usando uma determinada estrutura de dados

## Aplicações

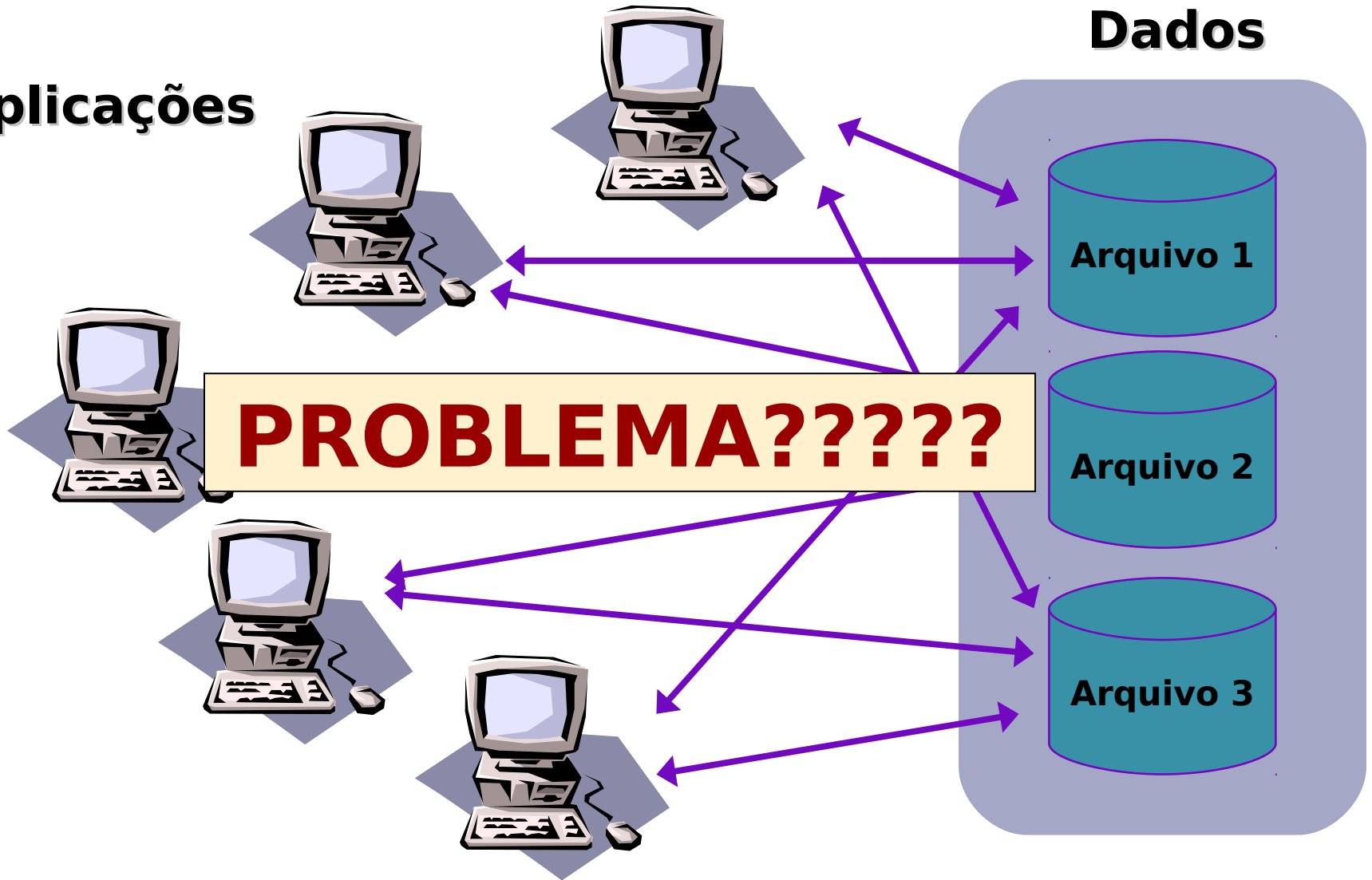


## Dados



**Aplicações**

**Dados**



# REDUNDÂNCIA

**Aplicação de  
Produção**



**Arquivos de Dados  
de Produção**



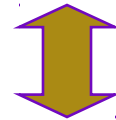
**Aplicação de  
Vendas**



**Arquivos de  
Dados de Vendas**



**Aplicação de  
Compras**



**Arquivos de Dados  
de Compras**



# REDUNDÂNCIA → INCONSISTÊNCIA

## Aplicação de Produção

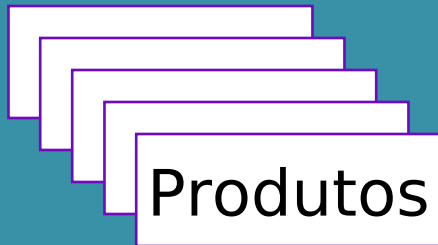


Insere:

Nome: Notebook  
NroSerie:1111111  
**Fabricante: Y**



Arquivos de Dados de Produção



## Aplicação de Vendas

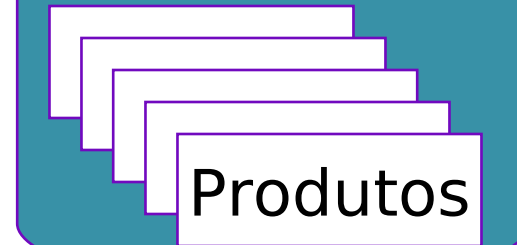


Insere:

Nome: Notebook  
NroSerie:1111111  
**Fabricante: X**

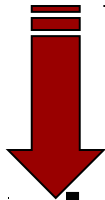


Arquivos de Dados de Vendas



# Consistência de Dados

- Dados em estado **inconsistente**



informações **incorretas** ou **contraditórias** são fornecidas aos usuários



# Consistência de Dados

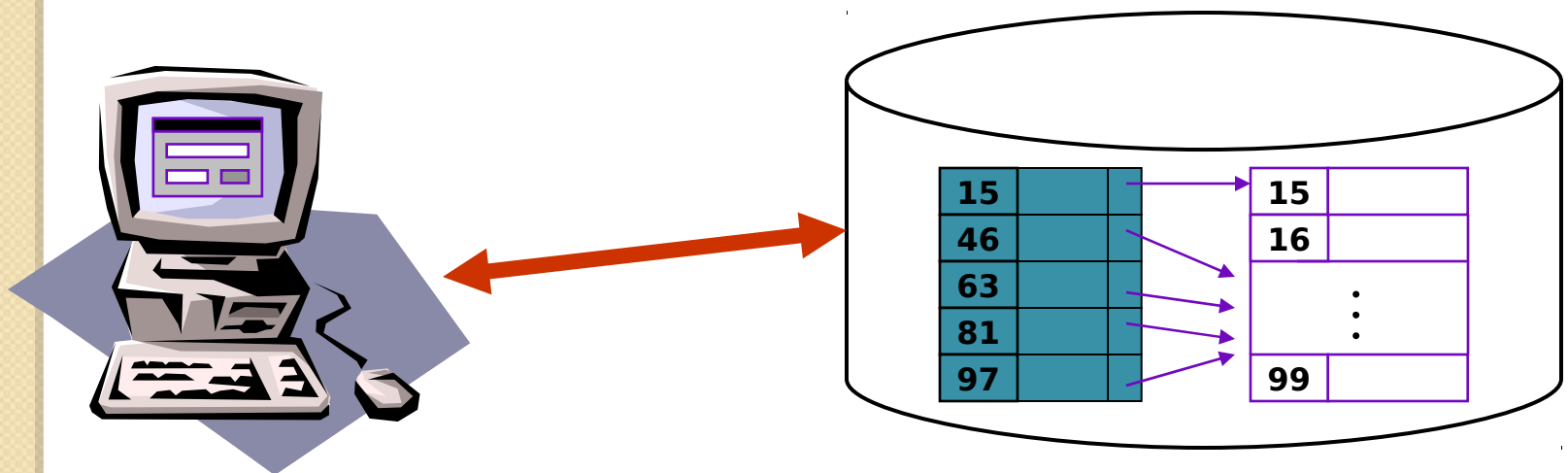
- **Consistência** é “estado ou caráter do que é coerente, do que tem solidez, veracidade, credibilidade, estabilidade, realidade”.
- Consistência: se determinada informação é replicada (redundância), seu valor é sempre o mesmo

# SIs baseados em arquivos

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

# Além disso...

- SIs baseados em arquivos  $\Rightarrow$  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  $\Rightarrow$  todos devem conhecer e manipular as mesmas estruturas
- E se houver uma alteração na estrutura de dados?

**TODOS OS PROGRAMAS TERÃO QUE SER ALTERADOS**

# Independência dos Dados

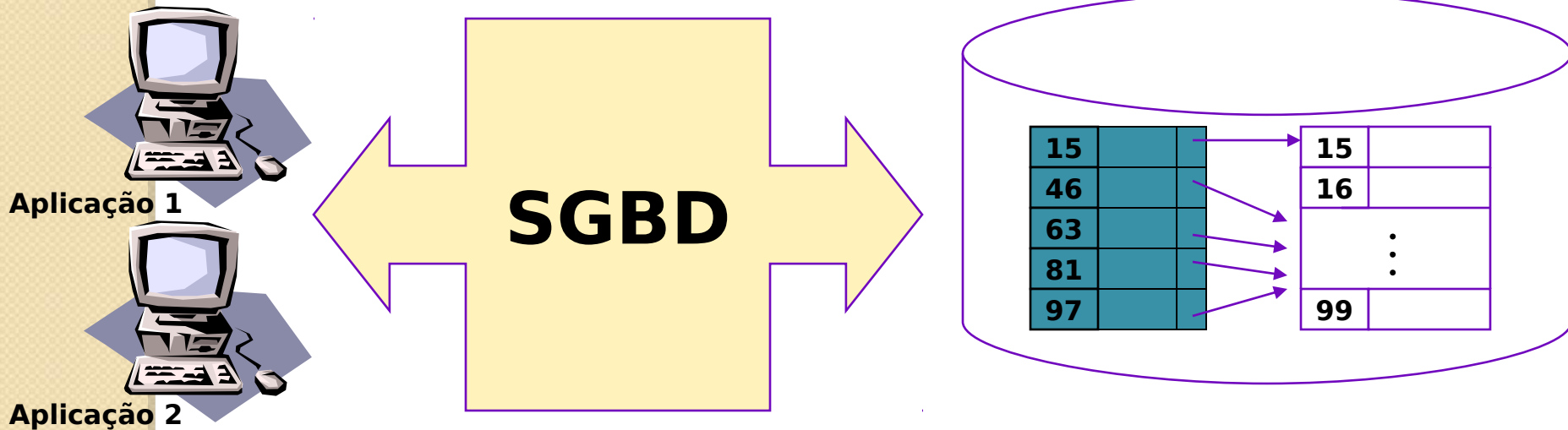
- Como tornar os programas **INDEPENDENTES** da estrutura de dados?

## CRIANDO UM SISTEMA QUE GERENCIE A ESTRUTURA



# Independência dos Dados

Sistema de Gerenciamento de Bases (ou Banco) de Dados  
**SGBD**



# SGBD

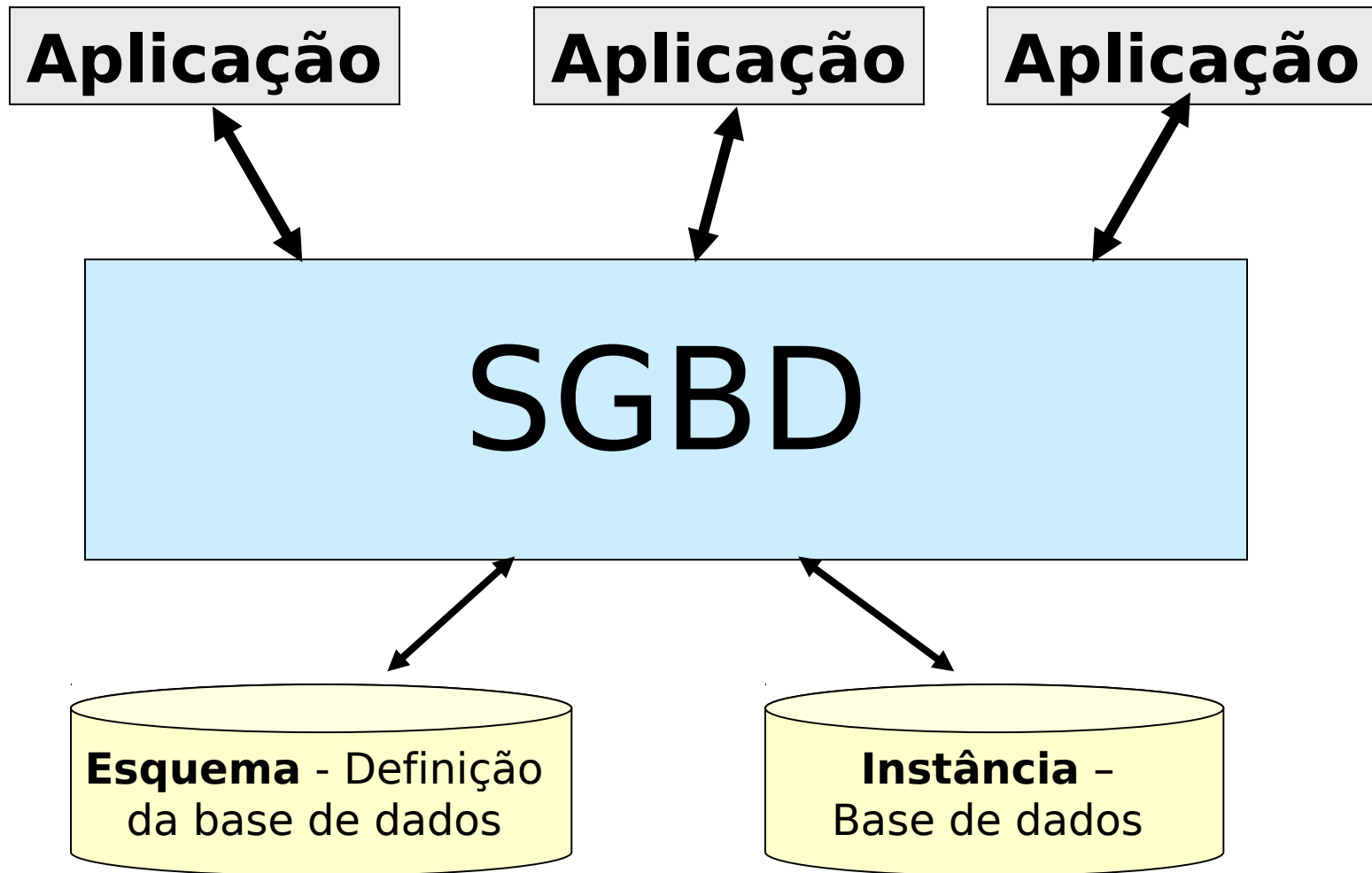
## Sistema de Gerenciamento de Bases de Dados

- conjunto de dados
  - base (banco) de dados
- conjunto de programas para acesso e manipulação dos dados

# SGBD

- Sistema de propósito geral
  - armazenar **grandes volumes** de dados
  - permitir busca e atualização dos dados
  - **eficiência**
- Manutenção de um conjunto lógico e organizado de dados
  - completamente autônomo em relação às aplicações



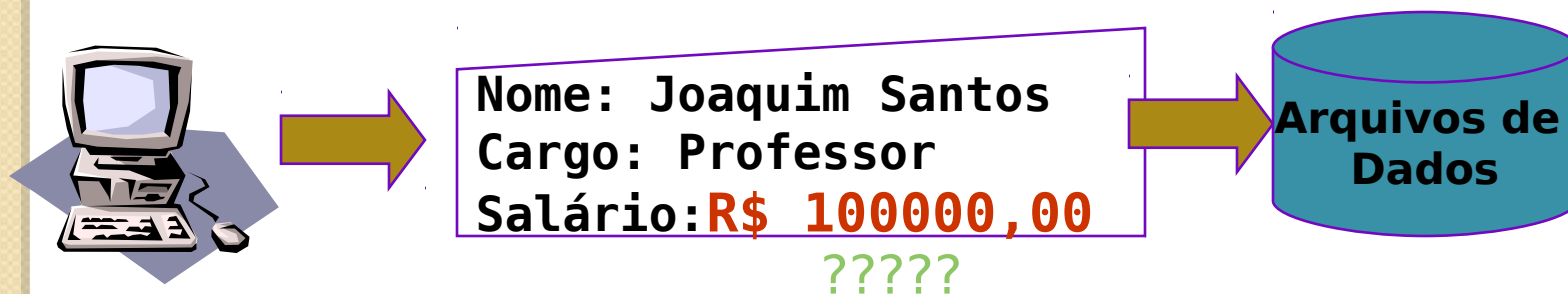


# SGBDs

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

# SGBDs

- Requisitos Fundamentais (cont):
  - Integridade
    - consistência
    - validade



**Restrições de Integridade!!!**

# SGBDs

- Requisitos Fundamentais (cont):
  - Recuperação / Tolerância a falhas
    - Transações atômicas
      - unidades lógicas de trabalho, em geral envolvendo várias operações
    - Registros de *Log*
    - *Backup*
  - Controle da concorrência
    - gerenciamento transações concorrentes

# Por que usar SGBDs?

- Vantagens:
  - armazenamento persistente de dados e estruturas de dados;
  - **INDEPENDÊNCIA DE DADOS;**
  - **CONSISTÊNCIA DE DADOS;**
  - acesso compartilhado (multiusuário e concorrente) à informação;
  - distribuição de informações

# Por que usar SGBDs?

- Vantagens:
  - reduz complexidade das aplicações
  - segurança
  - controle de acesso aos dados
  - *backup*
  - utilização de padrões

# Por que usar SGBDs?

- Desvantagens
  - Alto custo
  - Um sistema a mais a ser aprendido e **gerenciado**

# Componentes de um SGBD



- Os componentes funcionais do SGBD:
  - componentes de processamento de consultas
  - componentes de gerenciamento de armazenamento



# Componentes de um SGBD

- Conceitos importantes:
  - ***Data Definition Language*** (DDL)
    - conjunto de comandos para definição do esquema da base de dados
    - Exemplos em linguagem SQL
      - *create table*
      - *alter table*
      - *drop table*
  - Compilador/Interpretador DDL

# Componentes de um SGBD

- Conceitos importantes (cont.):
  - **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

# Componentes de um SGBD

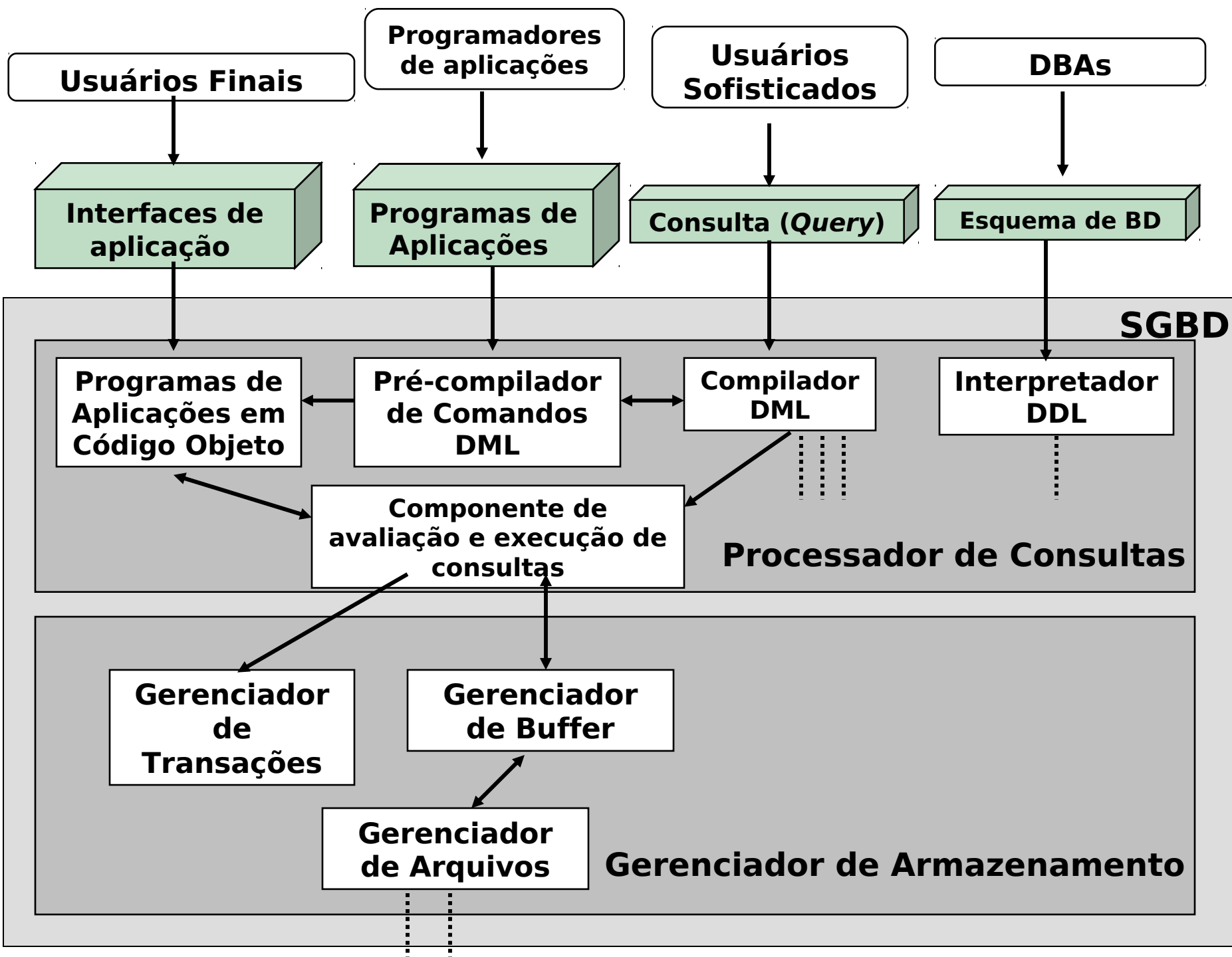
- Conceitos importantes (cont.):
  - ***Data Manipulation Language*** (DML)
    - recuperação (consulta)
    - inserção
    - remoção
    - modificação
    - DML viabiliza manipulação dos dados de maneira compatível com o modelo de dados

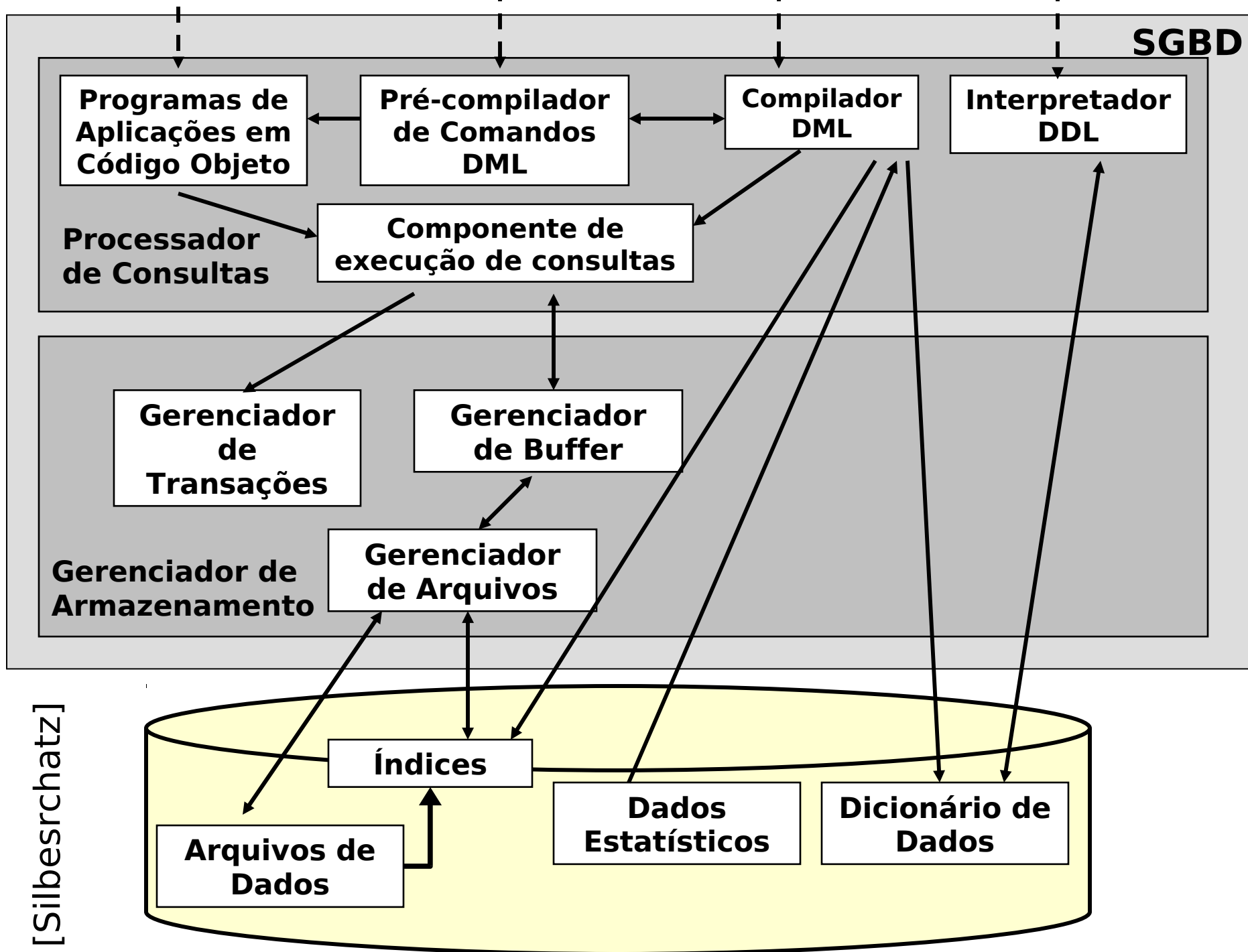
# Componentes de um SGBD

- Conceitos importantes (cont.):
  - ***Data Manipulation Language*** (DML)
    - Exemplos em linguagem SQL
      - *insert*
      - *select*
      - *delete*
      - *update*
      - ...

# Componentes de um SGBD

- Conceitos importantes (cont.):
  - Dois tipos de DML
    - **Procedural**: exige especificação de quais dados são necessários, e como obtê-los
      - requer uma sequência específica de operações a serem executadas
      - ex: álgebra relacional
    - **Não-Procedural (Declarativa)**: exige apenas especificação de quais dados são necessários, e não de como obtê-los
      - ex: SQL





# Tópicos da aula

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



# Cronologia

- Década de 1950:
  - Primeiros computadores.
  - Programação em linguagem de máquina
  - Surgimento de Sistemas Operacionais e Linguagens de Programação
  - Sistemas de arquivos com acesso não sequencial
    - 1959: Sistema RAMAC (IBM)

# Cronologia

- Década de 1960:
  - CODASYL
    - Conference on Data System Languages
  - Conceituação de SGBD e modelos de dados
  - Proposta do **Modelo de Rede**
  - Primeiro SGBD comercialmente disponível - **Modelo Hierárquico.**
    - IMS – IBM

# Cronologia

- Década de 1970:
  - Proposta do **Modelo Relacional**.
  - Surgimento de protótipos de **SGBDR**
    - INGRES (UC – Berkeley)
    - **Sistema R** (IBM)
  - Proposta do **Modelo Entidade - Relacionamento (MER)**

# Cronologia

- Década de 1980:
  - Primeiro SGBDR de grande porte disponível
    - DB2 – IBM
  - Surge SQL vinculada ao Sistema R
  - SQL torna-se padrão
  - **Modelos Orientados a Objetos**

# Cronologia

- Década de 1990 em diante:
  - SGBDs orientados a objetos
    - O2
    - ObjectStore
    - Objectivity/DB
    - Jasmine
    - ...
  - SGBDs objeto-relacionais
    - Oracle8, Oracle9, Oracle10g, Oracle 11g
    - PostgreSQL
    - Informix

# Sugestão de leitura

- Capítulos introdutórios dos livros citados na bibliografia básica da disciplina