

---

# **Banco de Dados**

## **Introdução**

DCC-UFLA

Prof. Denilson Alves Pereira

# Visão Geral

---

- Aplicações de banco de dados tradicionais
  - Armazenam informações textuais ou numéricas
- Bancos de dados multimídia
  - Armazenam imagens, audio e vídeo
- Sistemas de informações geográficas (GIS)
  - Armazenam e analisam mapas, dados sobre o clima e imagens de satélite
- Sistemas de data warehousing e de processamento analítico on-line (OLAP)
  - Extraem e analisam informações comerciais em grandes bancos de dados
  - Auxiliam em tomadas de decisão

# Visão Geral

---

- Tecnologias de tempo real e bancos de dados ativos
  - Controlam processos industriais e de manufatura
- Bancos de dados NoSQL
  - Manipulam grandes volumes de dados
  - Operam normalmente em clusters de computadores

# Conceitos Básicos

---

- **Dados** são fatos conhecidos que podem ser armazenados e que têm algum significado implícito.  
*Exemplo:* Uma coleção de nomes, endereços e telefones das pessoas que você conhece.
- **Banco de Dados** é uma coleção de dados relacionados com as seguintes propriedades implícitas:
  - Representa algum aspecto do mundo real (Mini-Mundo ou Universo de Discurso).
  - É uma coleção coerente de dados com algum significado inerente.
  - É projetado, construído e ‘povoado’ com dados para um propósito específico.

# Conceitos Básicos

---

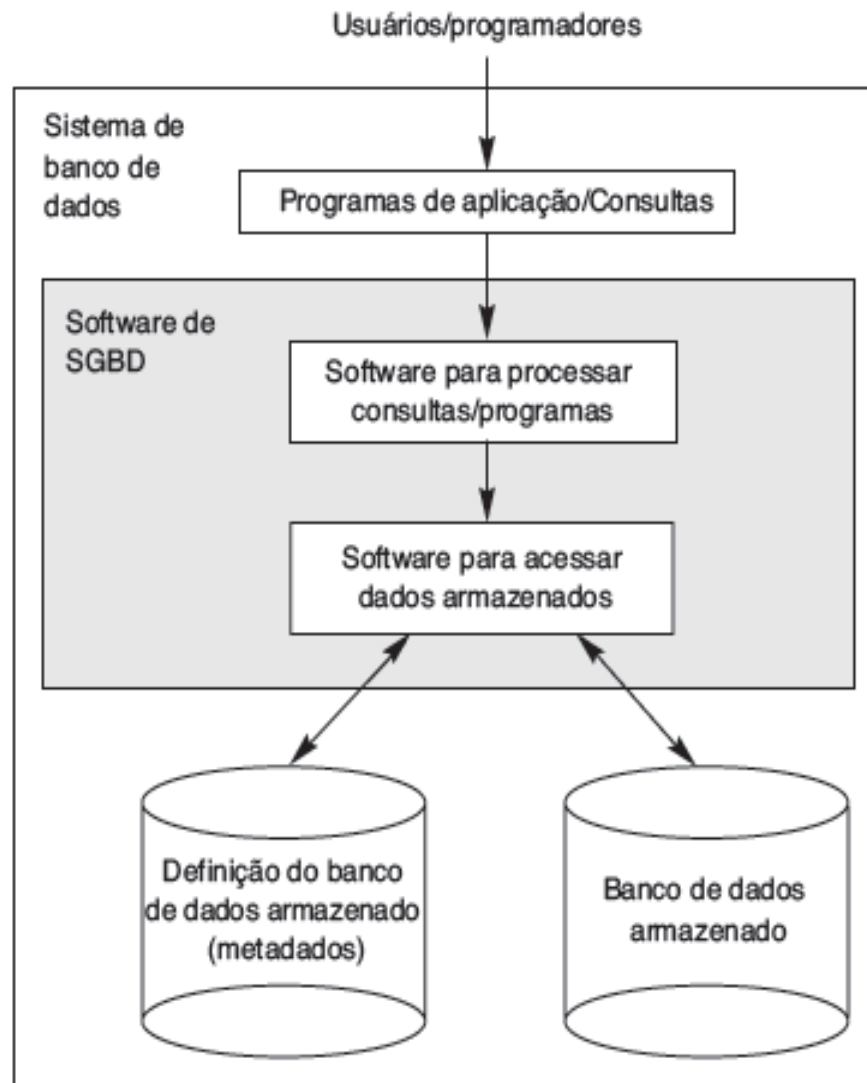
- Um banco de dados pode ser criado e mantido manualmente ou por máquina.
- Um banco de dados computadorizado pode ser criado e mantido por um grupo de programas de uma aplicação feitos especificamente para aquela tarefa ou por um sistema de gerência de banco de dados.
- ***Sistema de Gerência de Banco de Dados*** (SGBD ou DBMS) é uma coleção de programas que permite ao usuário criar e manter um banco de dados.

# Conceitos Básicos

---

- Um SGBD é um software de propósito geral que facilita os seguintes processos envolvendo bancos de dados de várias aplicações:
  - *Definir* - envolve especificar os tipos de dados, as estruturas e as restrições para os dados que serão armazenados.
  - *Construir* - é o processo de armazenamento dos dados em alguma mídia controlada pelo SGBD.
  - *Manipular* - envolve a aplicação de funções para recuperar informações específicas, modificar o banco de dados e gerar relatórios sobre os dados.
- ***Sistema de Banco de Dados*** é o conjunto de software para manipular um banco de dados juntamente com o banco de dados.

# Ambiente do Sistema de Banco de Dados



# Exemplo

Estudante	Nome	CódigoEstudante	Classe	DepartamentoPrincipal
	João	17	1	DCC
	José	8	2	DCC

Disciplina	Nome	CódigoDisciplina	Créditos	Departamento
	Introdução à Ciência da Computação	DCC1310	4	DCC
	Estrutura de Dados	DCC3320	4	DCC
	Matemática Discreta	MAT2410	3	MAT
	Banco de Dados	DCC3380	3	DCC

Turma	CódigoTurma	CódigoDisciplina	Semestre	Ano	Professor
	85	MAT2410	II	2011	King
	92	DCC1310	II	2011	Anderson
	102	DCC3320	I	2012	Knuth
	112	MAT2410	II	2012	Chang
	119	DCC1310	II	2012	Anderson
	135	DCC3380	II	2012	Stone

Histórico	CódigoEstudante	CódigoTurma	Nota
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PréRequisito	CódigoDisciplina	CódigoPréRequisito
	DCC3380	DCC3320
	DCC3380	MAT2410
	DCC3320	DCC1310



# Banco de Dados X Sistema de Arquivos Tradicional

---

- Sistema de Arquivos - cada usuário define e implementa os arquivos necessários para uma aplicação específica.
- Banco de Dados - uma única base de dados é definida e mantida para ser usada por vários usuários.
- Em um sistema de banco de dados:
  - As informações são armazenadas no Catálogo e são chamadas de metadados.
  - O catálogo é usado pelo SGBD e pode vir a ser consultado pelo usuário do banco de dados.
- Em processamento de arquivos tradicional:
  - A definição de dados faz parte do programa da aplicação.

# Banco de Dados X Sistema de Arquivos Tradicional

---

- Independência entre programas e dados e abstração de dados:
  - Arquivos - a estrutura do arquivo de dados está embutida nos programas que realizam o acesso aos dados.
  - Banco de Dados - os programas de acesso aos dados são independentes da estrutura de armazenamento dos dados.
  - Com o uso da orientação por objetos, as operações sobre os dados são definidas como parte do banco de dados, permitindo a independência entre programa e operações.
  - ***Abstração de dados*** é a característica que permite a independência entre programa e dados e a independência entre programa e operações.

# Banco de Dados X Sistema de Arquivos Tradicional

---

- Suporte a múltiplas visões dos dados:
  - Uma visão pode ser um subconjunto do banco de dados ou pode ser composta por dados virtuais que são derivados dos arquivos do banco de dados, mas não estão explicitamente armazenados.
- Compartilhamento de dados e processamento de transações multiusuário:
  - Um SGBD multiusuário deve permitir que vários usuários acessem o banco de dados ao mesmo tempo.
  - O acesso simultâneo é essencial se os dados de várias aplicações são integrados e mantidos em um único banco de dados.
  - Um SGBD deve possuir software de controle de concorrência.

# Usuários de um SGBD

---

- *Administradores do Banco de Dados (DBA)* - são os responsáveis pela administração do banco de dados, do SGBD e de seu software. Autorizam o acesso, coordenam e monitoram o uso e adquirem recursos de software e hardware quando necessário.
- *Projetistas do Banco de Dados* - são os responsáveis pela identificação dos dados a serem armazenados no banco de dados e pela escolha da estrutura apropriada para representar e armazenar esses dados.

# Usuários de um SGBD

---

- *Usuários Finais* - são as pessoas cujos trabalhos requerem acessar o banco de dados para fazer consultas, modificações e gerar relatórios. Tipos de usuário final:
  - *Casuais* - raramente acessam o banco de dados, mas podem precisar de diferentes informações em cada acesso.
  - *Iniciantes ou parametrizados* - seu trabalho envolve a realização de consultas e modificação do banco de dados constantemente. Eles usam consultas e modificações programadas e testadas.
  - *Especializados* - usuários completamente familiarizados com as facilidades do SGBD e que realizam requisitos complexos.
  - *Isolados* - mantêm um banco de dados pessoal usando pacotes prontos e fáceis de usar.

# Usuários de um SGBD

---

- *Analistas de Sistemas* - determinam os requisitos dos usuários finais e especificam o desenvolvimento de transações de acordo com os requisitos.
- *Programadores de Aplicações* - implementam as especificações como programas

# Trabalhadores por trás da cena

---

- *Projetistas e Implementadores de SGBDs* - são aqueles que projetam e implementam módulos e interfaces do SGBD.
- *Desenvolvedores de Ferramentas* - são aqueles que desenvolvem pacotes de software que facilitam o projeto e o uso do sistema de banco de dados e que auxiliam no aumento de performance.
- *Operadores e responsáveis pela manutenção* - são aqueles responsáveis pela execução e manutenção de software e hardware para o sistema de banco de dados.

# O que deve fornecer um SGBD?

---

- *Controle de Redundância* - problemas com dados redundantes: uma única alteração lógica deverá ser feita diversas vezes, espaço em disco desperdiçado e inconsistência de dados.
- *Restrição de acesso* - em um banco de dados utilizado por vários usuários, nem todos podem acessar todas as informações e nem todos podem realizar todas as operações sobre o banco de dados.
- *Armazenamento persistente para objetos e estruturas de dados de programas* - um objeto é persistente se ele continua existindo após o término da execução de um programa e pode ser acessado diretamente por um outro programa.



# O que deve fornecer um SGBD?

---

- *Inferência em banco de dados usando regras de dedução* - os sistemas de banco de dados que possuem essa capacidade são chamados Sistemas de Bancos de Dados Dedutivos.
- *Múltiplas interfaces de usuários* - os usuários do banco de dados possuem um variado nível de conhecimento técnico.
- *Representação de relacionamentos complexos entre os dados* - um SGBD deve representar os relacionamentos entre os dados e também recuperar e modificar dados relacionados de forma fácil e eficiente.

# O que deve fornecer um SGBD?

---

- *Garantia das restrições de integridade* - as restrições podem ser simples como as relacionadas ao tipo de dado, ou complexas como as que verificam se um registro em um arquivo está relacionado a registros em outros arquivos.
- *Backup e Recuperação* - um SGBD deve fornecer facilidades para recuperar os dados se ocorrer uma falha de hardware ou de software.

# Quando não usar um SGBD

---

- Para se usar um SGBD existe um aumento de custo em relação ao tradicional sistema de arquivos que se deve a:
  - Investimento inicial em software, hardware e treinamento.
  - Generalidade que um SGBD fornece para definir e processar os dados.
  - Fornecimento de segurança, controle de concorrência, recuperação e funções de integridade.
- Isso torna desejável o uso do sistema de arquivos nas seguintes situações:
  - O banco de dados e as aplicações são simples, bem definidas e a expectativa de mudança é pequena.
  - Requisitos de tempo-real de alguns programas.
  - Acesso de vários usuários aos dados não é requerido.

# Modelo de Dados

---

- A abordagem de banco de dados esconde detalhes de armazenamento dos dados que não são necessários para muitos usuários. O principal meio para fornecer esta abstração é o modelo de dados.
- Um ***modelo de dados*** é um conjunto de conceitos que podem ser usados para descrever as estruturas de um banco de dados.
- *A estrutura de um banco de dados* envolve os tipos de dados, os relacionamentos e as restrições sobre os dados.

# Categorias dos Modelos de Dados

---

- Pode-se categorizar um modelo de dados de acordo com o tipo de conceitos que ele fornece para descrever a estrutura do banco de dados.
- ***Conceitual (ou Alto-Nível)*** - fornece conceitos para descrever os dados como a maioria dos usuários o percebem. Esses conceitos são como:
  - Entidade - representa objetos e conceitos do mundo real. Ex.: ‘empregado’ e ‘projeto’.
  - Atributo - representa alguma propriedade de uma entidade. Ex.: ‘nome’ e ‘salário’ do empregado.
  - Relacionamento - representa a interação entre entidades. Ex.: ‘trabalha’ - um relacionamento entre empregado e projeto.Exemplos de modelo conceitual: Modelo ER e Modelo OO.

# Categorias dos Modelos de Dados

---

- **Lógico** (ou *Representacional*) - fornece conceitos que podem ser compreendidos pelos usuários finais, mas não está muito longe de como os dados são organizados no computador. Representam os dados usando a estrutura dos registros.

Exemplos: Modelos Relacional, Rede e Hierárquico.

- **Físico** (ou *Baixo-Nível*) - fornece conceitos para descrever os detalhes de como os dados estão armazenados no computador. Esses conceitos são como:
  - formato dos registros.
  - ordenação dos registros.
  - vias de acesso - são utilizadas para recuperar de forma eficiente os registros do banco de dados.

# Esquemas e Instâncias

- *Esquema* é a descrição textual ou gráfica de um banco de dados de acordo com um determinado modelo de dados.
- A representação como diagramas de um esquema de banco de dados é chamado *diagrama do esquema*.
- Exemplo de diagrama do esquema:

Estudante

Nome	CódigoEstudante	Classe	Departamer
------	-----------------	--------	------------

Disciplina

Nome	CódigoDisciplina	Créditos	Departamento
------	------------------	----------	--------------

Turma

CódigoTurma	CódigoDisciplina	Semestre	Ano
-------------	------------------	----------	-----

Histórico

CódigoEstudante	IdentificadorTurma	Nota
-----------------	--------------------	------

PréRequisito

CódigoDisciplina	CódigoPréRequisito
------------------	--------------------

# Esquemas e Instâncias

---

- Cada objeto de um esquema é chamado de *construção do esquema*.
  - Exemplo: Estudante, Disciplina.
- O diagrama mostra apenas alguns aspectos do esquema, tais como, nome do tipo de registro e dos itens dos dados e alguns tipos de restrições.
- Os dados em um banco de dados podem mudar frequentemente.
- ***Instância*** é o conjunto de dados armazenado em um banco de dados em um determinado instante do tempo (descreve o estado do banco de dados).



# Esquemas e Instâncias

---

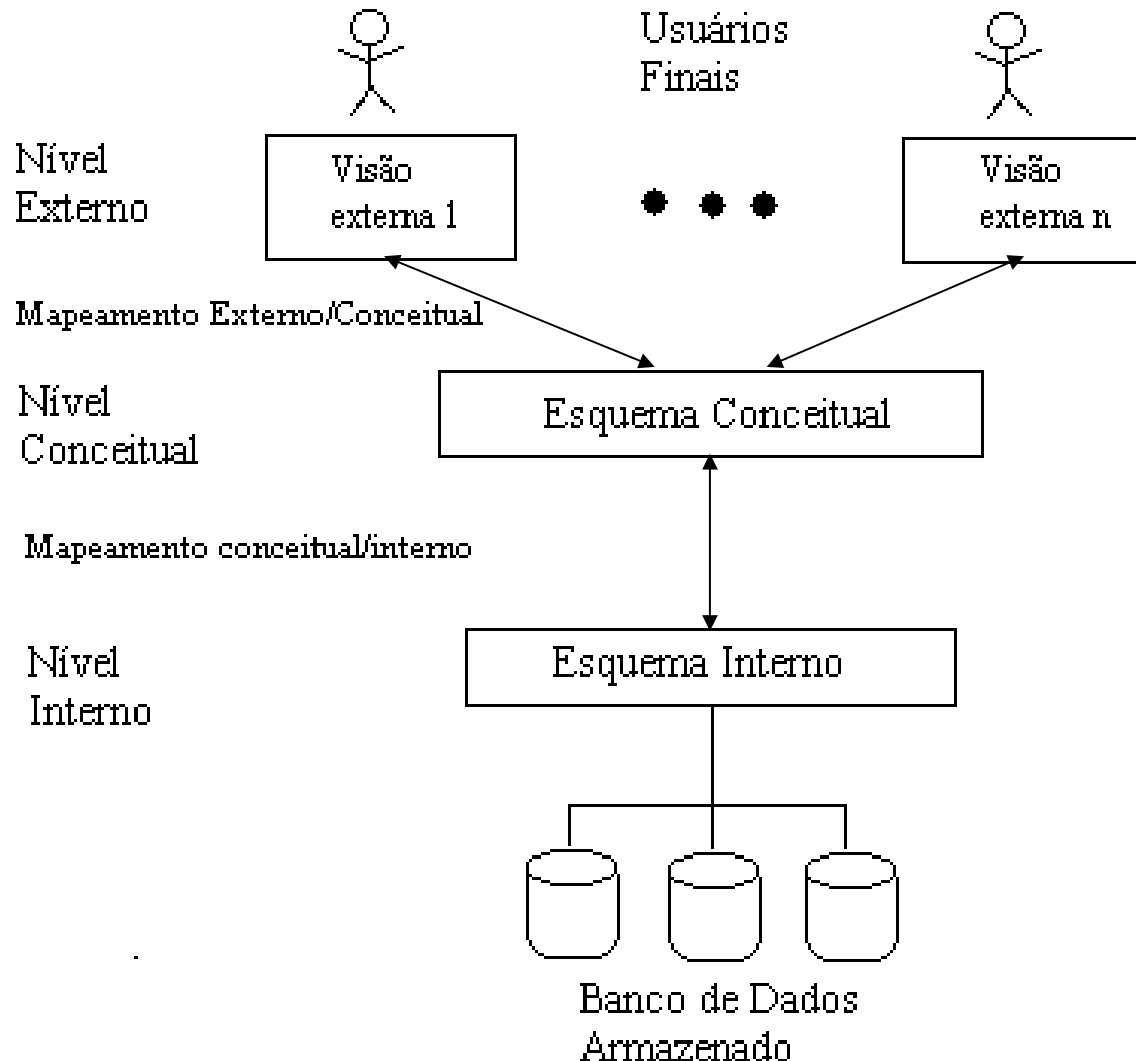
- Cada construção de esquema tem seu próprio conjunto de instâncias em um estado do banco de dados.
  - Exemplo: a construção Estudante contém um conjunto de registros de estudantes como suas instâncias.
- A definição de um novo banco de dados é apenas a especificação do esquema do banco de dados.
- O estado inicial de um banco de dados é obtido quando os primeiros dados são armazenados.
- O SGBD deve garantir que todo estado do banco de dados seja um estado válido.

# Arquitetura de Três Níveis (Três Esquemas)

---

- O objetivo da arquitetura de três níveis é separar as aplicações dos usuários e o banco de dados físico. Ela possui os seguintes níveis:
  - *Interno* - tem um esquema interno que descreve a estrutura física de armazenamento do banco de dados.
  - *Conceitual* - tem um esquema conceitual que descreve a estrutura completa do banco de dados para os usuários.
  - *Externo ou de Visão* - possui um conjunto de esquemas externos ou visões de usuários que descreve partes do banco de dados que um grupo particular de usuários pode estar interessado.

# Arquitetura de Três Níveis (Três Esquemas)



# Arquitetura de Três Níveis (Três Esquemas)

---

- Os três esquemas são apenas descrições de dados; o único local onde os dados realmente existem é no nível físico.
- Mapeamento - o SGBD deve transformar as requisições especificadas no esquema externo em requisições para o esquema conceitual e as do esquema conceitual em requisições para o esquema interno para ser processado sobre o banco de dados armazenado.

# Independência de Dados

---

- Independência de dados é a capacidade de mudar o esquema em um nível do sistema no banco de banco de dados sem ter que mudar o esquema do nível acima:
  - **Independência de dados lógica** - é a capacidade de mudar o esquema conceitual sem ter que mudar o esquema externo ou os programas da aplicação.
  - **Independência de dados física** - é a capacidade de mudar o esquema interno sem ter que mudar o esquema conceitual ou externo.

# Linguagens dos SGBDs

---

- Tipos de linguagens oferecidas por um SGBD:
  - *Linguagem de Definição de Dados (DDL)* - Em SGBDs que não possuem uma separação entre o nível conceitual e interno utiliza-se a DDL para definir os esquemas conceitual e interno. Naqueles que possuem uma separação, a DDL é utilizada para definir somente o esquema conceitual.
  - *Linguagem de Definição de Armazenamento (SDL)* - usada para especificar o esquema interno.
  - *Linguagem de Definição de Visões (VDL)* - especifica as visões dos usuários e o mapeamento para o esquema conceitual.
  - *Linguagem de Manipulação de Dados (DML)* - usada para consultar, inserir, remover e modificar dados do banco de dados.

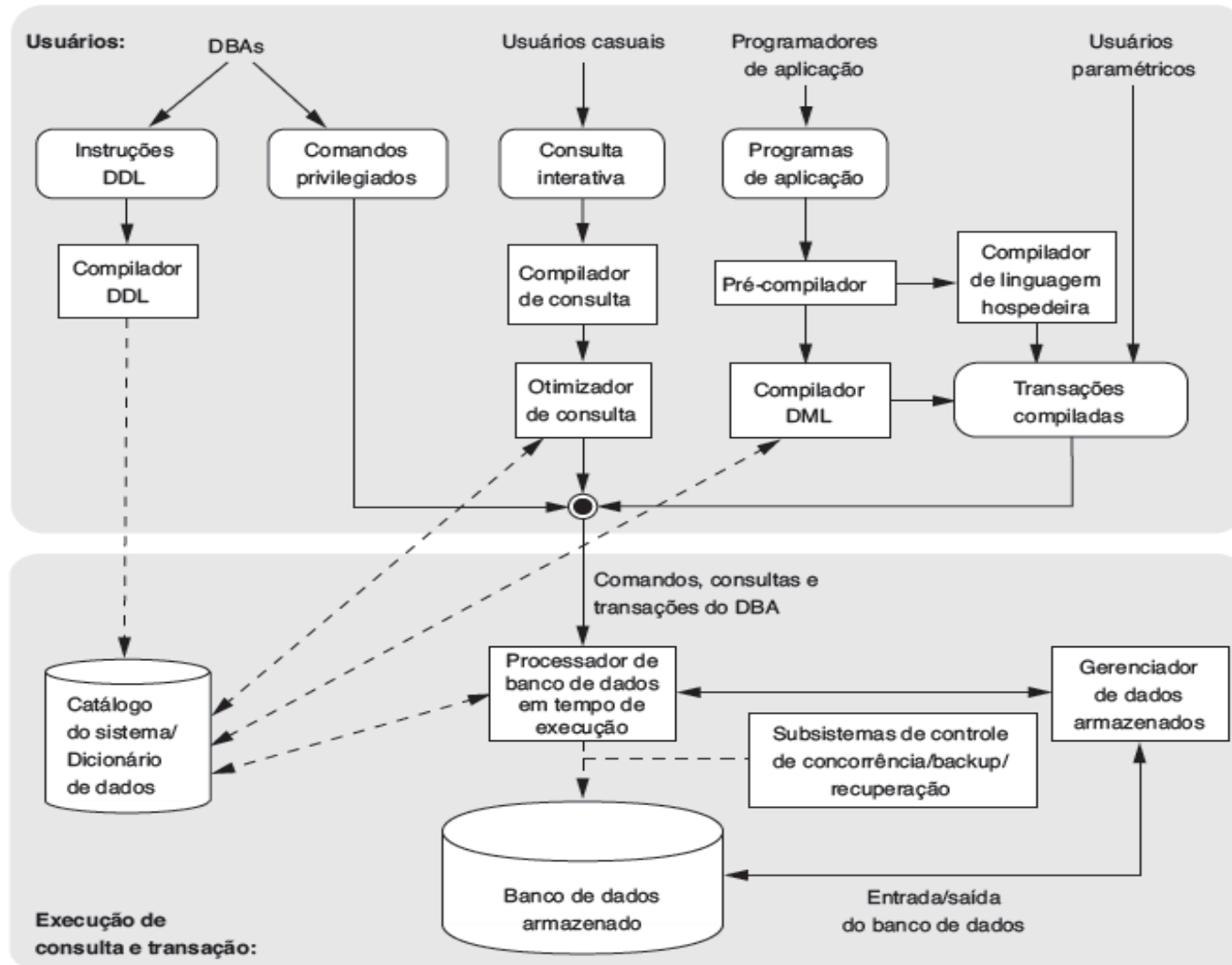
A linguagem de banco de dados relacional SQL representa uma combinação de DDL, SDL, VDL e DML.

# Linguagens dos SGBDs

---

- Tipos de Linguagens de Manipulação de Dados (DML):
  - *Não-procedural ou Alto-Nível* - requer do usuário a especificação de qual dado é necessário sem mostrar como obtê-lo.
    - As DMLs não procedurais podem ser executadas de forma interativa em um terminal ou podem ser embutidas em uma linguagem de programação.
  - *Procedural ou Baixo-Nível* - requer do usuário a especificação de qual dado é necessário e como obtê-lo.
    - As DMLs procedurais são embutidos em uma linguagem de programação.

# Módulos Componentes de um SGBD





# Módulos Componentes de um SGBD

---

- *Gerenciador do banco de dados* - Controla o acesso aos dados armazenados no disco.
- *Compilador da DDL* - Processa as definições de esquemas e armazena essas definições no catálogo do SGBD.
- *Processador do banco de dados em tempo de execução* - recebe as operações de recuperação e modificação e as executa sobre o banco de dados.
- *Compilador de consultas* - manuseia as consultas de alto nível que são entradas interativamente.

# Módulos Componentes de um SGBD

---

- *Pré-compilador* - extrai comandos DML de um programa de aplicação escritos em uma linguagem de programação hospedeira.
- *Compilador da DML* - recebe os comandos extraídos pelo pré-compilador e gera o código objeto para acessar o banco de dados.

O código objeto e restante do programa compilado são unidos formando uma transação.

# Utilitários

---

- *Carregador* - é utilizado para povoar o banco de dados com arquivos de dados existentes.
- *Backup* - cria uma cópia de backup do banco de dados.
- *Reorganizador de arquivos* - é utilizado para reorganizar um arquivo do banco de dados para melhorar a performance.
- *Monitor de performance* - monitora o uso do banco de dados e oferece estatísticas para o Administrador do banco de dados (DBA).

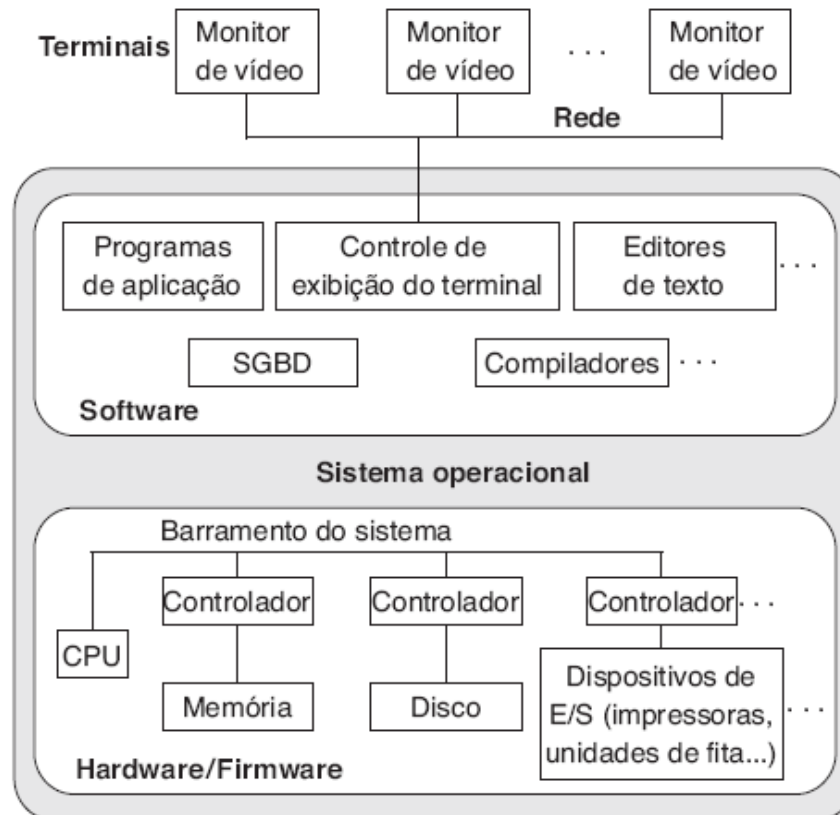
# Arquiteturas para SGBDs

---

- Centralizadas
- Cliente/Servidor básicas
- Cliente/Servidor de duas camadas
- Três camadas e n camadas para aplicações Web

# Arquiteturas Centralizadas

Toda a funcionalidade do SGBD, execução de programas de aplicação e processamento de interface do usuário são executadas em uma máquina, normalmente um mainframe



# Arquiteturas Cliente/Servidor Básicas

---

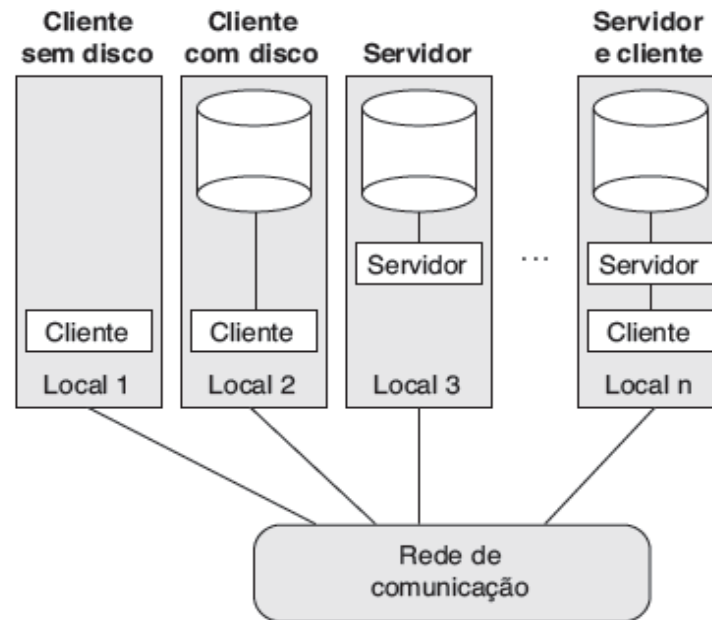
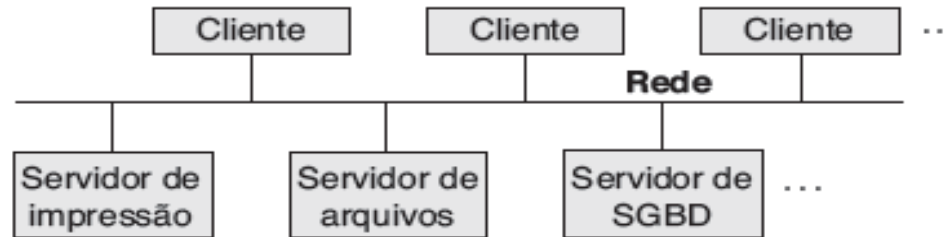
- Operam em redes de computadores, onde cada máquina assume o papel de servidor e/ou de cliente de algum recurso.
- Cliente
  - Máquina que oferece capacidades de interface com o usuário e processamento local
- Servidor
  - Sistema de hardware e software que oferece serviços às máquinas cliente, tais como acesso a arquivos, impressão e acesso a banco de dados

# Arquiteturas Cliente/Servidor Básicas

---

- Servidores típicos:
  - Servidor de arquivos: mantém os arquivos das máquinas cliente
  - Servidor de impressão: máquina conectada a várias impressoras para a qual as solicitações de impressão dos clientes são encaminhadas
  - Servidor Web: gerencia as solicitações de acesso à Web
  - Servidor de e-mail: gerencia as solicitações de correio
  - Servidor de banco de dados: fornece acesso a bancos de dados
- Máquinas cliente:
  - Possuem interfaces apropriadas para utilizar os servidores

# Arquiteturas Cliente/Servidor Básicas





# Arquiteturas Cliente/Servidor de duas Camadas

---

- Os componentes de software são distribuídos por dois sistemas: cliente e servidor
- Servidor:
  - Nos SGBD relacionais, fornecem as funcionalidades de consulta e de processamento de transações
- Cliente:
  - Executam os programas da interface com o usuário e os programas de aplicação

# Arquiteturas Cliente/Servidor de duas Camadas

---

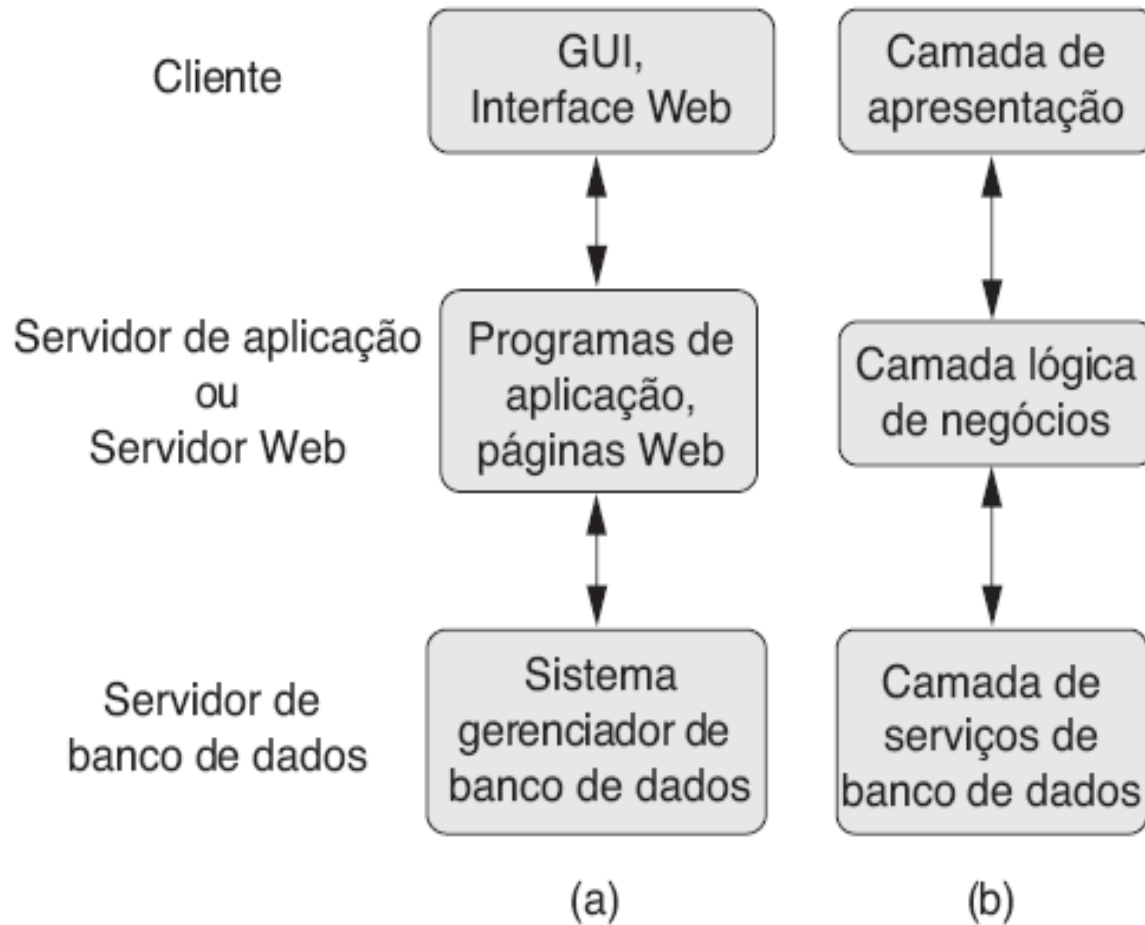
- Padrão ODBC (*Open Database Connectivity*)
  - Oferece uma interface de programação de aplicações (API)
  - Permite que os programas do cliente acessem o SGBD
- Padrão JDBC (*Java Database Connectivity*)
  - Conectividade para programas em Java
  - Permite que programas em Java acessem um ou mais SGBDs por meio de uma interface padrão

# Arquiteturas de Três camadas e n camadas para aplicações Web

---

- Três camadas
  - Acrescenta uma camada intermediária entre o cliente e o servidor de banco de dados
  - Atua com servidor de aplicação ou servidor Web
  - Executa programas de aplicação e armazena regras de negócio
- N camadas
  - Divide a camada intermediária em outros componentes mais detalhados
  - A camada de lógica de negócios é dividida em várias camadas

# Arquiteturas de Três camadas e n camadas para aplicações Web



# Critérios de Classificação dos SGBDs

---

- Modelo de dados: relacional, hierárquico, rede, orientado a objeto, objeto-relacional e XML.
- Número de usuários: mono-usuário e multiusuário.
- Número de sites:
  - Centralizado - os dados estão armazenados em um único computador.
  - Distribuídos (SGBDD) - o banco de dados e os softwares dos SGBDs estão em vários locais conectados por uma rede de computadores.
    - SGBDD Homogêneos - usam o mesmo SGBD nos vários sites.
    - SGBD Federados - os SGBDs participantes são heterogêneos e possuem uma forma de ligação entre eles, e cada um tem um grau de autonomia local.

# Referência Bibliográfica Básica

---

ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de Bancos de Dados. Pearson Education, 6ª edição, 2011. ISBN-978-85-7936-085-5