

# [MAC0439] Laboratório de Bancos de Dados

## Aula 2

### Introdução aos Sistemas de Bancos de Dados (Continuação)

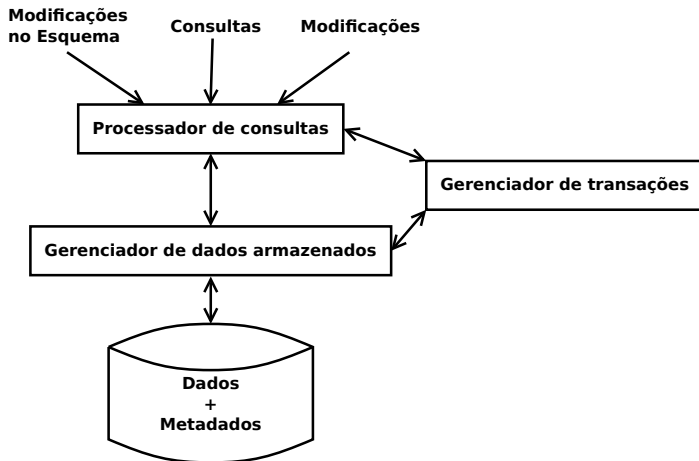
Kelly Rosa Braghetto

DCC-IME-USP

14 de agosto de 2015

# Linguagens Usadas na Interação com os SGBDs

# Principais componentes de um SGBD (uma visão simplificada)



# Linguagens de um SGBD

## Linguagem de Definição dos Dados (DDL – *Data Definition Language*)

- ▶ Em SGBDs que usam modelos de dados onde a separação entre os níveis não é muito específica, a DDL é usada pelo DBA (administrador do BD) e pelos projetistas do BD para definir os esquemas conceitual e interno para o BD
- ▶ Em SGBDs que mantêm uma clara separação entre os esquemas conceitual e físico, a DDL é usada somente para especificar o esquema conceitual
  - ▶ Uma outra linguagem – a **linguagem de definição de armazenamento** (STL – *Storage Definition Language*) é usada na especificação do esquema interno

# Linguagens de um SGBD

## Linguagem de Manipulação de Dados (DML – *Data Manipulation Language*)

- ▶ É a linguagem oferecida pelo SGBD para a manipulação de dados
- ▶ Permite a definição de operações de manipulação típicas, como recuperação, inserção, alteração e remoção de dados
- ▶ Pode ser de dois tipos:
  - ▶ **de baixo nível** (ou **procedural**) – é embutida em uma linguagem de programação hospedeira de uso geral. Geralmente recupera registros individuais do BD e os processa separadamente.
  - ▶ **de alto nível** (ou **de consulta**) – são declarativas (ou seja, especificam *quais* registros recuperar e não *como* recuperá-los). Podem recuperar vários registros em uma única instrução.

# Linguagens de um SGBD

## Linguagem de Definição de Visões (VDL – *View Definition Language*)

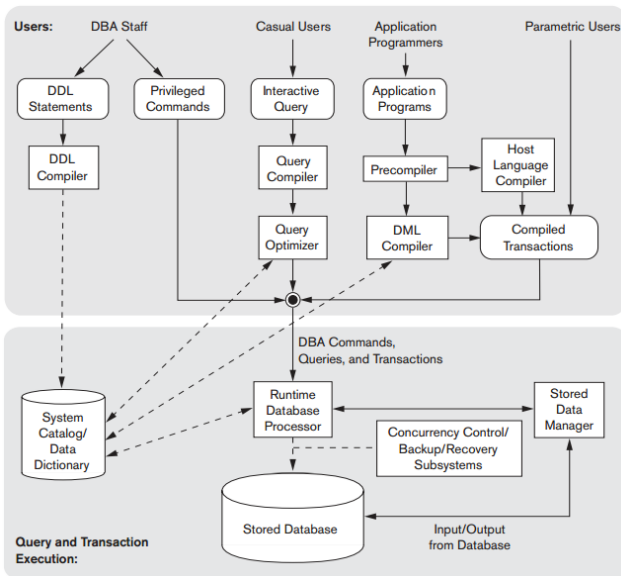
- ▶ A VDL permite especificar visões de usuários e seus mapeamentos para o esquema conceitual

⇒ Nos SGBDs atuais, muitas vezes a DDL, a DML e a VDL são “integradas” em uma única linguagem.

Exemplo: a linguagem padrão para bancos de dados relacionais  
**SQL** (*Structured Query Language*)

# Arquiteturas dos SGBDs Convencionais

# Principais componentes de um SGBD





# Arquitetura de SGBDs centralizada (1960 – 1970)

## Contexto:

- ▶ Arquitetura usada nos primeiros SGBDs
- ▶ Computadores mainframes: ofereciam o processamento principal para todas as funções do sistema (programas de aplicação e interface com usuário + SGBD)
- ▶ Usuários acessavam esses sistemas por meio de terminais “burros” (com capacidades de exibição, mas sem poder de processamento)

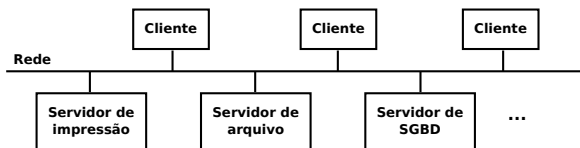
## Principal característica:

- ▶ O SGBD era um sistema **monolítico** (altamente integrado), executado em uma única máquina

# Arquiteturas cliente/servidor básicas

## Características:

- ▶ Desenvolvidas para lidar com ambientes onde há um grande número de PCs, estações de trabalho, impressoras, servidores de e-mail, servidores Web, servidores de bancos de dados, etc., conectados por rede
- ▶ Ideia geral: definir **servidores especializados**, com funcionalidades específicas
- ▶ **Máquinas clientes** oferecem aos usuários as interfaces apropriadas para utilizar esses servidores



Arquitetura cliente/servidor lógica em duas camadas

## Arquiteturas cliente/servidor de duas camadas para SGBDs

- ▶ Componentes inicialmente movidos para o lado do cliente: interface com o usuário e os programas de aplicação
- ▶ SQL (linguagem padrão para os SGBDs): criou um ponto de divisão lógico entre o cliente e o servidor
  - ▶ servidor  $\Rightarrow$  **servidor de consulta** ou **servidor de transação** (funcionalidades relacionadas ao processamento da SQL)
  - ▶ quando necessita acessar o SGBD, o programa cliente estabelece uma conexão com o servidor
- ▶ **Padrão Conectividade de Banco de Dados Aberta** (ODBC – *Open Database Connectivity*) – API que permite que programas clientes se comuniquem com um SGBD de forma padronizada
  - ▶ A maioria dos “fabricantes” de SGBDs disponibiliza *drivers* ODBC dos seus sistemas

# Arquiteturas cliente/servidor de duas camadas para SGBDs

## Variação usada em SGBDs de objetos

- ▶ Cliente se ocupa também de funcionalidades do SGBD tais como: otimização de consulta, controle de concorrência e recuperação entre vários servidores; estruturação de objetos complexos; interação do SGBD com as linguagens de programação; etc.
- ▶ Assim, o servidor se ocupa, por exemplo, das funcionalidades diretamente relacionadas à manipulação das páginas no disco, controle de concorrência local e recuperação.

⇒ Nessa abordagem, a interação cliente/servidor é mais acoplada (feita internamente pelos módulos do BD, e não pelos usuários/programadores).

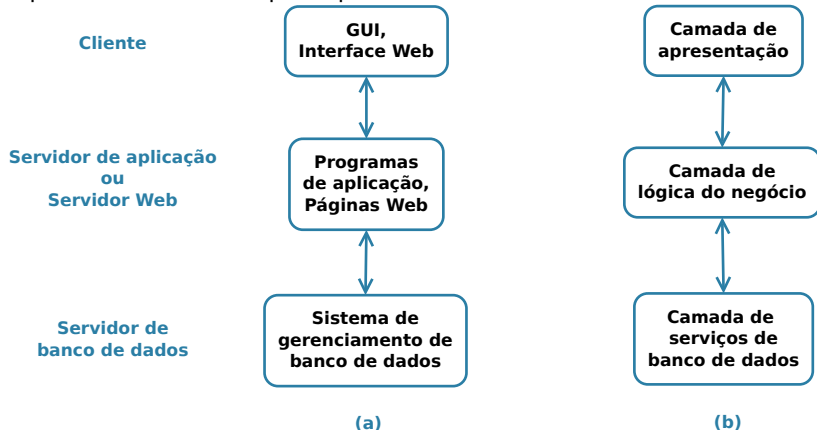
Nesse contexto, o servidor é chamado de **servidor de dados**.

## Arquiteturas de 3 camadas para aplicações Web (1990)

- ▶ Usam uma camada intermediária entre o cliente e o servidor, chamada de **servidor de aplicação** ou **servidor Web**.
- ▶ Esse servidor executa programas de aplicação ou armazena regras de negócio (procedimentos ou restrições) que são usados para acessar os dados do servidor de banco de dados.
- ▶ Podem também adicionar um nível superior de segurança, verificando as credenciais de um cliente antes de encaminhar uma requisição ao banco de dados.
- ▶ Clientes contêm interfaces gráficas de usuário e alguma regras de negócio adicionais específicas da aplicação.

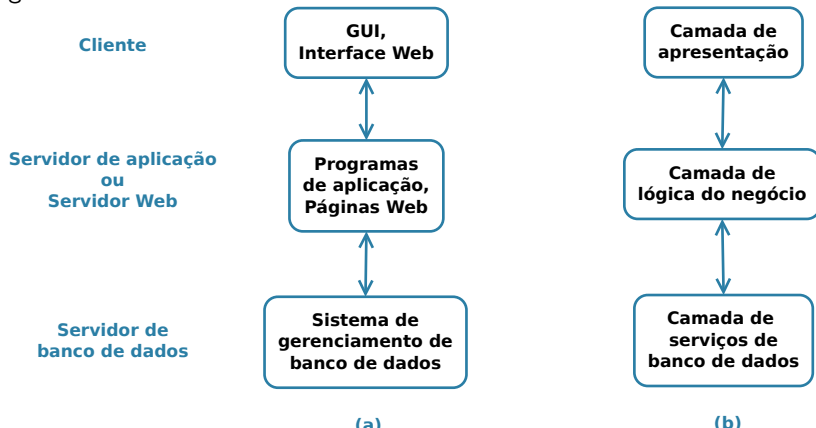
# Arquiteturas de 3 camadas para aplicações Web

**Figura (a)** – O servidor intermediário aceita requisições de clientes, as processa e envia as consultas e comandos para o servidor de BD. Depois, repassa a resposta enviada pelo servidor de BD para o cliente, que pode reprocessá-la e filtrá-la para apresentar aos usuários.



## Arquiteturas de 3 camadas para aplicações Web

**Figura (b)** – A camada de apresentação mostra informações para usuários e permite a entrada de dados. A camada de lógica do negócio trata de regras intermediárias e restrições antes que os dados sejam passados de/para o SGBD. A camada de baixo inclui todos os serviços de gerenciamento de dados.



# Classificação dos SGBDs



# Classificação dos Sistemas de Gerenciamento de Banco de Dados

## Critérios normalmente utilizados:

- ▶ **Propósito** – uso geral ou uso específico
- ▶ **Modelo de dados usado** – modelo relacional, de objetos, multidimensional, XML, de grafo, chave-valor, colunar, etc.
- ▶ **Número de usuários** – monousuário (admite apenas um usuário de cada vez) ou multiusuário (admitem múltiplos usuários simultaneamente)
- ▶ **Custo e “Liberdade”** – código aberto/gratuito (como o PostgreSQL e o MySQL/MariaDB), código fechado/pago (como MS SQL Server, Sybase, Oracle)

# Classificação dos Sistemas de Gerenciamento de Banco de Dados

## Critérios normalmente utilizados (continuação):

- ▶ **Número de locais** sobre os quais o BD está distribuído
  - ▶ SGBD **centralizado** – os dados estão armazenados em um único computador
  - ▶ SGBD **distribuído** – pode ter os dados e/ou o software do SGBD distribuídos por vários locais, conectados por uma rede. Pode ser homogêneo (quando todos os locais possui um mesmo software de SGBD) ou heterogêneo (quando cada local possui um software diferente).
  - ▶ SGBD **federado** – middleware que acessa vários bancos de dados autônomos, armazenados em SGBDs heterogêneos.
- ▶ **Tipos de caminhos de acesso** para armazenar arquivos – ex.: estrutura de arquivos invertidos

## Referências Bibliográficas

- ▶ *Sistemas de Bancos de Dados* (6ª edição), Elmasri e Navathe. Pearson, 2010.  
Capítulos 1 e 2
- ▶ *Database Systems – the complete book* (2ª edição), Garcia-Molina, Ullman e Widom. Prentice Hall, 2009.  
Capítulo 1
- ▶ *Introdução a Sistemas de Bancos de Dados* (8ª edição), Date. Campus, 2004.  
Capítulos 1 e 2

# Cenas dos próximos capítulos...

## Revisão sobre:

- ▶ Projeto conceitual de BDs e o modelo entidade-relacionamento
- ▶ Projeto lógico de BDs e o modelo relacional