

[MAC0426] Sistemas de Bancos de Dados
[IBI5013] Bancos de Dados para Bioinformática
Aula 2
Conceitos e Arquitetura do Sistema de Banco de
Dados

Kelly Rosa Braghetto

DCC-IME-USP

16 de março de 2017

Modelos de Dados

Abstração de dados

Oferecer abstração de dados é uma característica fundamental dos bancos de dados, ocultando detalhes sobre a organização e armazenamento dos dados (detalhes esses que são desnecessários para a maioria dos usuários de BDs)

A abstração é feita por meio de modelos de dados:

- ▶ **Modelo de dados** – é um conjunto de conceitos que podem ser usados para descrever a *estrutura* de um banco de dados
 - ▶ Modelos também podem incluir *operações* básicas para a recuperação e atualização de dados do banco
- ▶ **Estrutura de um banco de dados** – define os tipos de dados, relacionamentos e restrições que se aplicam aos dados

Um “parênteses” sobre a especificação de operações em BDs

- ▶ Além das operações básicas, agora é comum que modelos de dados ofereçam mecanismos para a especificação do aspecto dinâmico (= comportamento) de uma aplicação de BD
- ▶ O projetista do BD pode especificar operações definidas pelos usuários, permitidas sobre os objetos do BD
- ▶ Esses mecanismos são particularmente importantes para os modelos de dados de objetos, mas também vêm sendo incorporados aos modelos de dados mais tradicionais

Tendência atual: combinar as atividades de projeto de banco de dados às de projeto de software

Categorias de modelos de dados

- ▶ **Modelos de dados conceituais** (ou **de alto nível**) – possuem conceitos que descrevem os dados como os usuários os percebem
- ▶ **Modelos de dados físicos** (ou **de baixo nível**) – possuem conceitos que descrevem os detalhes de como os dados estão armazenados no computador, com informações como o formato de um registro, as ordenações dos registros e os caminhos de acesso ¹
- ▶ **Modelos de dados representativos** (ou **de implementação**) – estão a um meio termo dos dois anteriores, pois podem ser entendidos pelos usuários finais, ao mesmo tempo em que não estão tão distantes da forma como os dados são organizados e armazenados no computador

¹Estrutura que torna eficiente a busca por registros em um BD particular

Modelos de dados conceituais

Possuem conceitos tais como:

- ▶ **Entidade:** representa um objeto ou conceito do mundo real (ex.: aluno, disciplina, turma, ...)
- ▶ **Atributo:** representa alguma propriedade que ajuda a descrever uma entidade (ex.: nome do aluno, número USP, código da disciplina, ...)
- ▶ **Relacionamento:** mostra uma associação entre duas ou mais entidades (ex.: aluno pertence a uma turma)

⇒ Exemplo de modelo conceitual bastante popular: **modelo entidade-relacionamento**

Modelos de dados representativos

- ▶ Os modelos representativos são os mais usados nos SGBDs comerciais tradicionais
- ▶ Exemplos: o **modelo de dados relacional**, o **modelo de redes** e o **modelo hierárquico**
Esses modelos são denominados de **modelos de dados baseados em registros** (pois mostram os dados usando estruturas de registros)
- ▶ O **modelo de dados de objeto** pode ser considerado como um modelo representativo de mais alto nível, mas também é bastante usado como modelo conceitual (especialmente na área de Engenharia de Software).

Descrição do BD × BD de fato

Esquema do BD

⇒ É a descrição do BD

- ▶ É definido durante a fase de projeto e espera-se que não seja alterado com frequência
- ▶ Geralmente, é representado por um **diagrama**, que mostra somente alguns aspectos do esquema (nome dos registros e itens de dados, alguns tipos de restrições)

Descrição do BD × BD de fato

Estado do BD (ou *snapshot*, ou *conjunto de instâncias*)

⇒ São os dados no BD em um determinado momento no tempo

- ▶ Toda inserção ou remoção de um registro, ou alteração do valor de um item de dado gera uma mudança de estado no BD
- ▶ O SGBD é parcialmente responsável por garantir que cada novo estado do BD é um estado válido (= que respeita a estrutura e as restrições definidas no esquema)
- ▶ O SGBD armazena em seu catálogo as descrições das construções do esquema e suas restrições (= metadados) e, dessa forma, pode acessá-los sempre que necessário

Exemplo de diagrama de esquema

ALUNO

Nome	Numero_aluno	Tipo_aluno	Curso
------	--------------	------------	-------

DISCIPLINA

Nome_disciplina	Numero_disciplina	Creditos	Departamento
-----------------	-------------------	----------	--------------

PRE_REQUISITO

Numero_disciplina	Numero_pre_requisito
-------------------	----------------------

TURMA

ID_turma	Numero_disciplina	Semestre	Ano	Professor
----------	-------------------	----------	-----	-----------

HISTORICO_ESCOLAR

Numero_aluno	ID_turma	Nota
--------------	----------	------

Um “parênteses” sobre a evolução do esquema de um BD

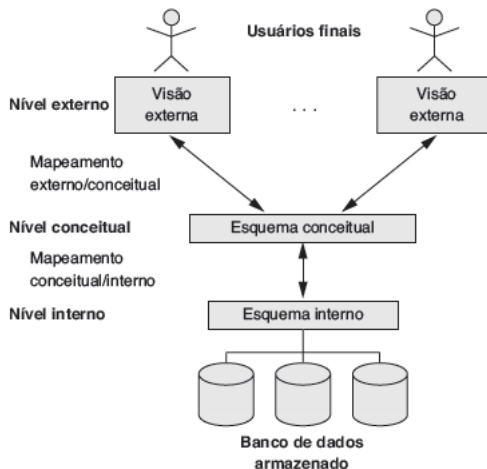
- ▶ Embora o esquema de um BD não deva mudar frequentemente, não é raro que ele sofra alterações ao longo do tempo
- ▶ Isso é chamado de **evolução do esquema**
- ▶ A maioria dos SGBDs modernos possui operações que permitem que um esquema seja evoluído ao mesmo tempo em que o BD está em funcionamento

Arquitetura de Três Esquemas (Arquitetura ANSI-SPARC)

ANSI: American National Standards Institute

SPARC: Standards Planning And Requirements Committee

Objetivo: separar as aplicações de usuários do banco de dados físico



Arquitetura de Três Esquemas

Essa arquitetura garante para um BD:

- ▶ Natureza autodescritiva (esquema armazenado em catálogo)
- ▶ Isolamento entre programas e dados, e abstração de dados
- ▶ Suporte a visões múltiplas dos dados

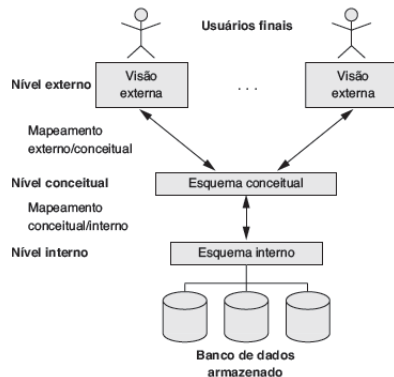
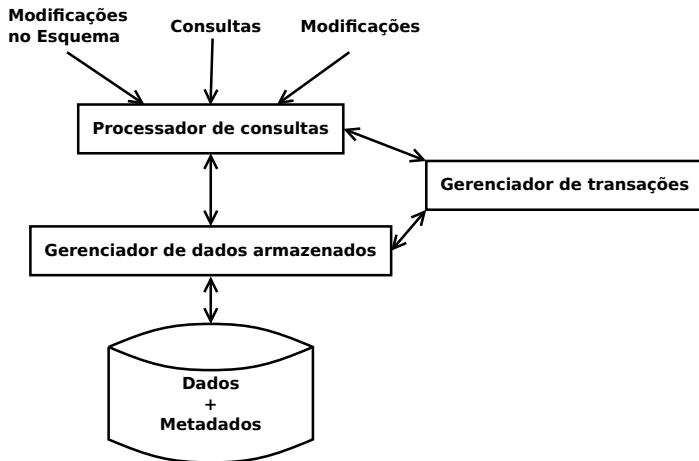


Figura 2.2

A arquitetura de três esquemas.

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*)

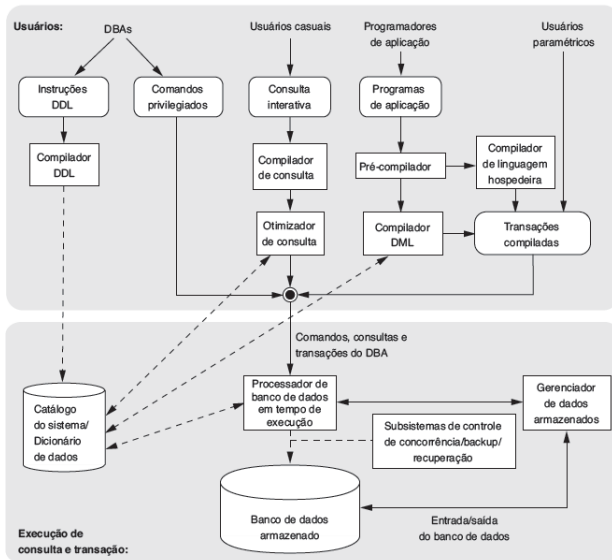
- ▶ A VDL permite especificar visões de usuários e seus mapeamentos para o esquema conceitual

⇒ 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)

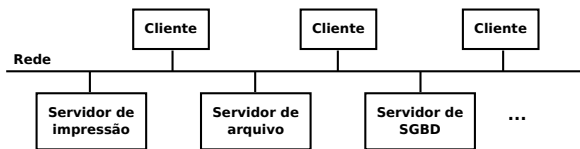
Principal característica:

- ▶ 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 SGBDRs): 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).

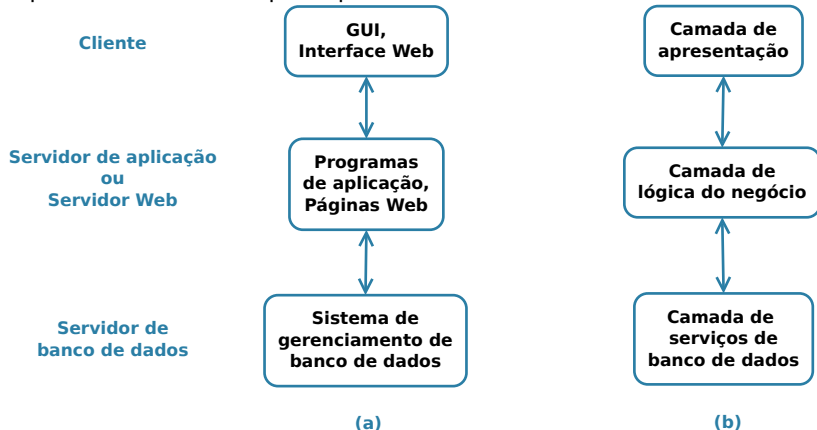
Nesse contexto, o servidor é chamado de **servidor de dados**.

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.

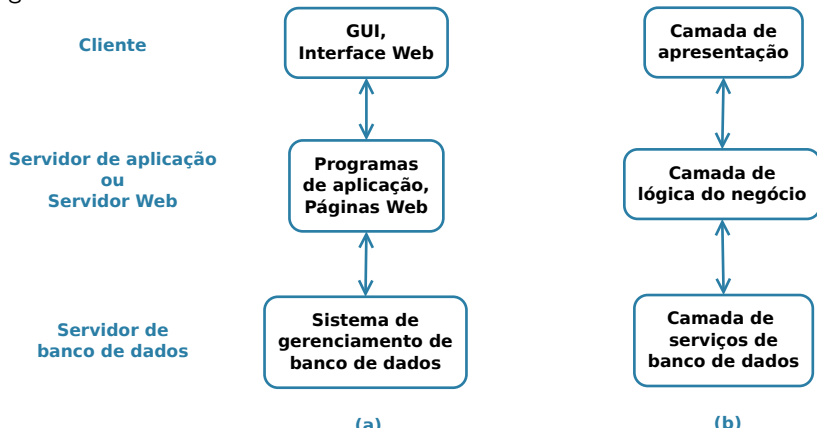
Arquiteturas de 3 camadas para aplicações Web

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.
Capítulo 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...

Projeto conceitual de BDs

- ▶ O modelo entidade-relacionamento