

[MAC0313]

Introdução aos Sistemas de Bancos de Dados

Aula 2

Conceitos e Arquitetura dos Sistemas de Bancos de Dados

Kelly Rosa Braghetto
(kellyrb@ime.usp.br)

DCC-IME-USP

04 de agosto de 2016

Modelos de Dados

Abstração de dados

Oferecer abstração de dados é uma característica fundamental dos SGBDs, 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

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

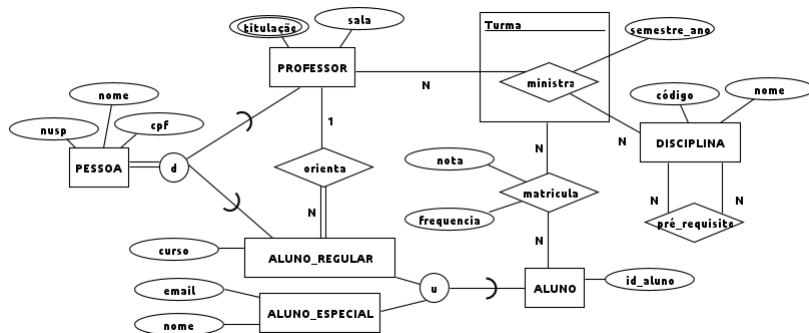
Possuem conceitos tais como:

- ▶ **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 conceituais

Exemplo de diagrama no 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).

Modelos de dados representativos

Exemplo de diagrama no modelo relacional

ALUNO

Nome	<u>Numero_aluno</u>	Tipo_aluno	Curso
------	---------------------	------------	-------

DISCIPLINA

Nome_disciplina	<u>Numero_disciplina</u>	Creditos	Departamento
-----------------	--------------------------	----------	--------------

PRE_REQUISITO

<u>Numero_disciplina</u>	<u>Numero_pre_requisito</u>
--------------------------	-----------------------------

TURMA

<u>ID_turma</u>	Numero_disciplina	Semestre	Ano	Professor
-----------------	-------------------	----------	-----	-----------

HISTORICO_ESCOLAR

<u>Numero_aluno</u>	<u>ID_turma</u>	Nota
---------------------	-----------------	------

Descrição do BD $\times$ BD de fato

Esquema do BD

$\Rightarrow$ 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 $\times$ BD de fato

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

$\Rightarrow$ 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 estado de BD

ALUNO

Nome	Numero_aluno	Tipo_aluno	Curso
Silva	17	1	CC
Braga	8	2	CC

DISCIPLINA

Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciência da computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC

TURMA

Identificacao_turma	Numero_disciplina	Semestre	Ano	Professor
85	MAT2410	Segundo	07	Kleber
92	CC1310	Segundo	07	Anderson
102	CC3320	Primeiro	08	Carlos
112	MAT2410	Segundo	08	Chang
119	CC1310	Segundo	08	Anderson
135	CC3380	Segundo	08	Santos

HISTORICO_ESCOLAR

Numero_aluno	Identificacao_turma	Nota
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PRE_REQUISITO

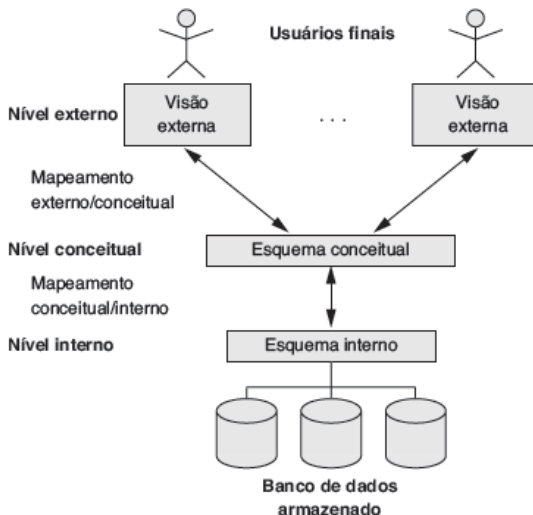
Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310

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

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

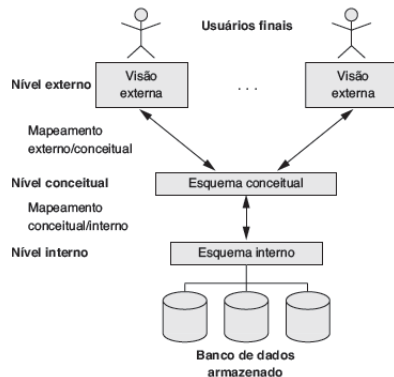
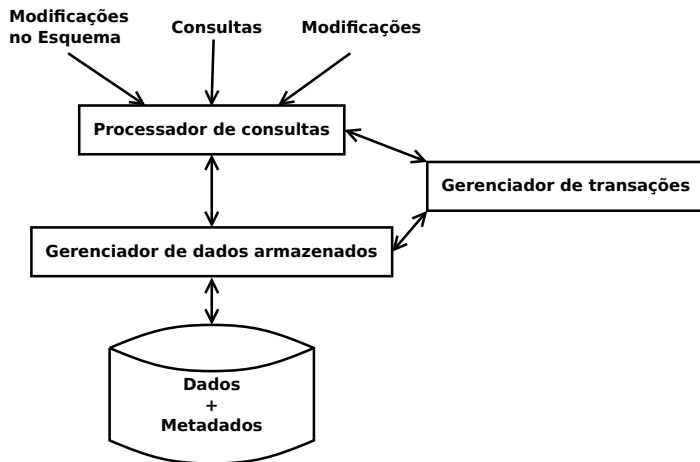


Figura 2.2

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

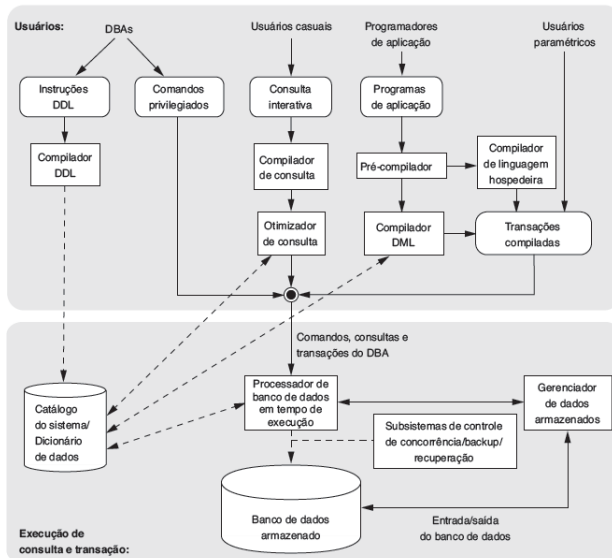
Linguagem de Definição de Visões (VDL – *View Definition Language*)

- ▶ 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*)

Principais componentes de um SGBD



Classificação dos SGBDs

Classificação dos SGBDs

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 (aditem 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 SGBDs

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