

[MAC0426] Sistemas de Bancos de Dados  
[IBI5013] Bancos de Dados para Bioinformática  
Aula 1  
Introdução aos Sistemas de Bancos de Dados

Kelly Rosa Braghetto

DCC-IME-USP

23 de fevereiro de 2016

# Bancos de Dados – Introdução

O que é um banco de dados?

Você sabe citar alguns exemplos?

# O que é um **banco de dados**?

- ▶ **Banco de dados** – coleção de dados relacionados
- ▶ **Dados** – fatos conhecidos que podem ser registrados e que possuem significado implícito

Problema: essa definição é genérica demais!

# Propriedades implícitas de um banco de dados (BD)

1. Representar (geralmente!) algum aspecto do mundo real = **minimundo** ou **UoD** (Universo de Discurso)  
As mudanças no minimundo são refletidas no BD
2. Ser uma **coleção lógica e coerente de dados** com algum **significado inerente**  
Uma coleção “aleatória” de dados não é um BD!
3. Ser projetado, construído e povoado com dados que possuem um objetivo específico  
Um BD deve possuir um grupo provável de usuários e algumas aplicações pré-concebidas, nas quais esses usuários estão interessados

# Propriedades implícitas de um banco de dados (BD)

**Resumindo:** um BD possui alguma **fonte** (de onde os dados são derivados), algum grau de **interação com eventos do mundo real** e um **público** que está ativamente interessado no conteúdo do BD

## Outras características:

- ▶ BDs têm complexidade e tamanho variáveis
- ▶ BDs podem ser informatizados ou mantidos manualmente

## Exemplos:

- ▶ Uma agenda de telefones e endereços de seus contatos
- ▶ O catálogo com as informações do acervo de uma biblioteca
- ▶ Os dados de imposto de renda da Receita Federal

# Exemplo da dimensão que um BD pode assumir

## Facebook (dados de abril de 2014)

- ▶ *Data warehouse* com mais de 300 PB (petabytes)
- ▶ Diariamente, cerca de 600 TB (terabytes) de novos dados
- ▶ Mais de 1 bilhão de usuários ativos
- ▶ Grande variedade de aplicações: desde do tradicional processamento em lotes até a análise de grafos (redes), aprendizagem de máquina e análise interativa em tempo real.
- ▶ Em 2013, o quantidade de dados armazenados no *data warehouse* **triplicou**

1 petabyte = 1.000 terabytes = 1 quadrilhão de bytes  
(~ 210.000 DVDs)

Fonte:

<https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/>

# Softwares para a manutenção de bancos de dados

Um BD informatizado pode ser criado e mantido por:

- ▶ um grupo de programas de aplicação (criados especificamente para essa tarefa)

**ou**

- ▶ um **Sistema de Gerenciamento de Banco de Dados (SGBD)**

Sistema de software de propósito geral que facilita o processo de **definição, construção, manipulação e compartilhamento** de BDs entre vários usuários e aplicações



# Sistema de Gerenciamento de Banco de Dados (SGBD)

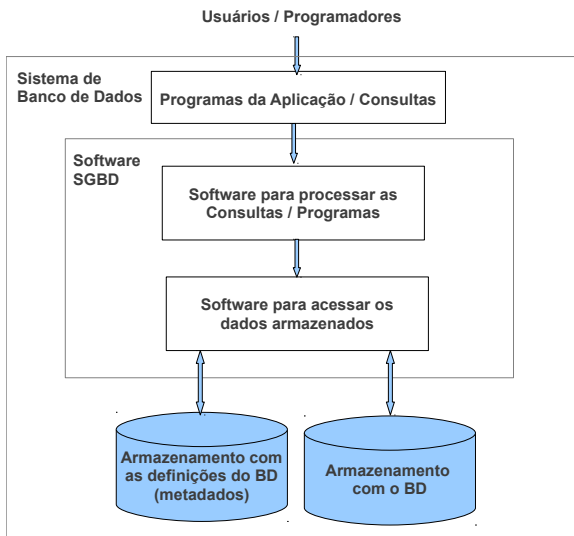
## Apoia o ciclo de vida de um BD:

- ▶ **Definir um BD**  $\Rightarrow$  especificar os tipos, as estruturas e as restrições para os dados que serão armazenados no BD
- ▶ **Construir um BD**  $\Rightarrow$  gravar os dados em algum meio de armazenamento (controlado pelo SGBD)
- ▶ **Manipular um BD**  $\Rightarrow$  realizar funções como consultas ao BD para recuperar dados específicos, atualizar o BD para refletir mudanças no minimundo, etc.
- ▶ **Compartilhar um BD**  $\Rightarrow$  permitir que múltiplos usuários e programas acessem-no simultaneamente

## Outras funções importantes de um SGBD

- ▶ **Proteger os dados** – contra falhas de hardware ou software e contra acessos não autorizados ou maliciosos
- ▶ **Manter os dados por um longo período de tempo** – permitindo que o sistema evolua acompanhando as mudanças dos requisitos ao longo do tempo

# Sistema de Banco de Dados = Banco de Dados + SGBD



## O acesso a um BD

Um **programa de aplicação** é um programa que acessa um banco de dados enviando **consultas** ou **transações** para o SGBD.

- ▶ **Consulta** – comando que recupera dados do BD
- ▶ **Transação** – comando que lê ou escreve dados do/no BD

# Exemplo de BD: informações de alunos e disciplinas

## 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

# Banco de dados × processamento de arquivos tradicional

## Banco de dados × processamento de arquivos tradicional

- ▶ No **processamento de arquivos tradicional**: cada usuário define e implementa os arquivos necessários para uma aplicação específica (como parte da aplicação)

**Problema: possível redundância na definição e armazenamento de dados** ⇒ desperdício de espaço e trabalho redundante na manutenção de dados comuns a mais de uma aplicação

- ▶ Nos **bancos de dados**: um único repositório de dados é mantido

**Uma vez definido, o repositório passa a ser acessado por diversos usuários e aplicações**

# Outras características importantes de BDs mantidos em SGBDs tradicionais

## Natureza autodescritiva

- ▶ BDs são mantidos com uma descrição completa de sua estrutura e restrições (metadados)
- ▶ Os metadados são armazenados no catálogo do SGBD, e são usados tanto pelo SGBD quanto por usuários do BD
- ▶ O SGBD precisa dos metadados porque ele trabalha com qualquer banco de dados (ou seja, **é de propósito geral**) :))
- ▶ Já no processamento de arquivos tradicional, a definição da estrutura dos dados está no código do programa de aplicação  
⇒ esses programas trabalham com um banco de dados específico :(



# Outras características importantes de BDs mantidos em SGBDs tradicionais

## Exemplo de metadados

### RELACOES

Nome_relacao	Numero_de_colunas
ALUNO	4
DISCIPLINA	4
TURMA	5
HISTORICO_ESCOLAR	3
PRE_REQUISITO	2

### COLUNAS

Nome_coluna	Tipo_de_dado	Pertence_a_relacao
Nome	Caractere (30)	ALUNO
Numero_aluno	Caractere (4)	ALUNO
Tipo_aluno	Inteiro (1)	ALUNO
Curso	Tipo_curso	ALUNO
Nome_disciplina	Caractere (10)	DISCIPLINA
Numero_disciplina	XXXXNNNN	DISCIPLINA
....	....	....
....	....	....
....	....	....
Numero_pre_requisito	XXXXNNNN	PRE-REQUISITO

## Outras características importantes de BDs mantidos em SGBDs tradicionais

### Isolamento entre programas e dados (por meio de **abstração de dados**)

- ▶ Um SGBD oferece uma representação conceitual dos dados que não inclui muitos detalhes sobre como eles são armazenados fisicamente
- ▶ A inclusão de um novo item de dado na estrutura no BD não implica na necessidade de alteração dos programas de aplicação que o acessam via SGBD; os programas continuarão funcionando corretamente :))
- ▶ Já no processamento de arquivos tradicional, qualquer mudança na estrutura de um arquivo implica na necessidade de se alterar todos os programas de aplicação que acessam esse arquivo :(

# Outras características importantes de BDs mantidos em SGBDs tradicionais

Exemplo: formato de armazenamento interno para um registro de ALUNO

## RELAÇOES

Nome_relacao	Numero_de_colunas
ALUNO	4
DISCIPLINA	4
TURMA	5
HISTORICO_ESCOLAR	3
PRE_REQUISITO	2

## COLUMNAS

Nome_coluna	Tipo_de_dado	Pertence_a_relacao
Nome	Caractere (30)	ALUNO
Numero_aluno	Caractere (4)	ALUNO
Tipo_aluno	Inteiro (1)	ALUNO
Curso	Tipo_curso	ALUNO
Nome_disciplina	Caractere (10)	DISCIPLINA
Numero_disciplina	XXXXNNNN	DISCIPLINA
....	....	....
....	....	....
....	....	....
Numero_pre_requisito	XXXXNNNN	PRE-REQUISITO

Nome do item de dados	Posicionamento inicial no registro	Tamanho em caracteres (bytes)
Nome	1	30
Numero_aluno	31	4
Tipo_aluno	35	1
Curso	36	4

# Outras características importantes de BDs mantidos em SGBDs tradicionais

## Suporte a visões múltiplas dos dados

- ▶ Um BD costuma ter muitos usuários
- ▶ Cada usuário pode exigir uma **visão** diferente do BD
- ▶ Uma visão pode ser um subconjunto do BD ou conter dados que são derivados dos dados armazenados. Exemplos:
  - ▶ Acesso apenas ao histórico escolar de um aluno
  - ▶ Acesso aos pré-requisitos de cada disciplina na qual um aluno se inscreveu
- ▶ Um SGBD oferece facilidades para definir múltiplas visões

# Outras características importantes de BDs mantidos em SGBDs tradicionais

## Compartilhamento de dados e processamento de transações multiusuários

- ▶ Num SGBD, múltiplos usuários podem acessar um mesmo BD ao mesmo tempo
- ▶ Isso pode ocorrer com frequência, já que o dado para múltiplas aplicações é integrado e mantido num único BD
- ▶ O SGBD faz controle de concorrência, para garantir que o resultado de tentativas de atualizações simultâneas sobre um mesmo dado resultem em algo correto

# Pessoas que interagem com os BDs e seus SGBDs

- ▶ Administrador de banco de dados (DBA – *database administrator*)
- ▶ Projetista de banco de dados
- ▶ Usuários finais
  - ▶ casuais – usam linguagens de consultas
  - ▶ paramétricos – usam transações programadas
  - ▶ sofisticados – engenheiros, cientistas, analistas de negócio, ...
  - ▶ isolados – mantêm BDs pessoais, usando pacotes prontos
- ▶ Analistas de sistemas e programadores

## Vantagens do uso de um SGBD

1. **Controle de redundâncias** – para evitar *duplicação de esforço, desperdício de espaço de armazenamento e inconsistência*
2. **Restrição do acesso não autorizado** – por meio da criação de usuários (protegidos por senha) com diferentes tipos de permissão de acesso
3. **Armazenamento persistente para objetos de programas e estruturas de dados** – mas há um problema: incompatibilidade entre as estruturas de dados oferecidas pelo BD e as existentes nas linguagens de programação

## Vantagens do uso de um SGBD (Cont.)

4. **Estruturas de armazenamento e técnicas de busca para o processamento eficiente de consultas** – por meio da criação de índices e da manutenção de *caches* em memória principal
5. **Mecanismos de *backup* e recuperação** – para garantir a integridade dos dados no caso de falhas
6. **Múltiplas interfaces de usuário** – para apoiar os diferentes perfis de usuários que interagem com os BDs



## Vantagens do uso de um SGBD (Cont.)

7. **Capacidade de representação de relacionamentos complexos entre dados**
8. **Imposição de restrições de integridade** – como: de tipo, de integridade referencial, de singularidade (chave)
9. **Possibilidade de deduzir dados e executar ações por meio de regras**

## Vantagens do uso de um SGBD (Cont.)

### Implicações adicionais:

1. Potencial para garantir padrões
2. Redução no tempo de desenvolvimento de aplicações
3. Flexibilidade
4. Disponibilidade de informações atualizadas
5. Economias de escala

# Breve história das aplicações de banco de dados

## Primórdios (de meados da década de 60 até os anos 80)

- ▶ não havia (suficientemente) *abstração de dados e independência entre dados e programas*
- ▶ o armazenamento físico era planejado para beneficiar determinados tipos de consultas, mas não oferecia flexibilidade e eficiência para acessos de por meio de outras consultas
- ▶ mudanças no BD para acomodar novos requisitos eram trabalhosas
- ▶ acesso aos dados apenas por meio de interfaces de linguagens de programação
- ▶ usava computadores *mainframes* (grandes e caros)
- ▶ 3 paradigmas: *modelo hierárquico, modelo de redes e sistemas de arquivos invertidos*

# Breve história das aplicações de banco de dados

BDs Relacionais: introduzido no final da década de 70, se desenvolvendo nos anos 80

- ▶ separação do armazenamento físico dos dados de sua representação conceitual
- ▶ base matemática para representação e consulta dos dados
- ▶ linguagens de consulta de alto nível
- ▶ flexibilidade para novas consultas e reorganização do banco de dados
- ▶ tipo de bancos de dados dominante para aplicações tradicionais (até hoje!)
- ▶ existem tanto em grandes servidores quanto em computadores pessoais

# Breve história das aplicações de banco de dados

## BDs Orientados a Objetos: final da década de 80

- ▶ motivados pelo surgimento das linguagens de programação OO e pela necessidade de armazenar e compartilhar objetos complexos
- ▶ complexidade do modelo e falta de padronização fizeram com que fossem pouco usados
- ▶ hoje estão incorporados ao SGBDs relacionais (os chamados SGBDs *objeto-relacionais*)

# Breve história das aplicações de banco de dados

XML, para intercâmbio de dados na Web (introduzido no final dos anos 90)

- ▶ partes das informações nas páginas Web de e-commerce são frequentemente extraídas de SGBDs
- ▶ técnicas foram desenvolvidas para possibilitar a troca de dados na Web
- ▶ a XML foi o primeiro formato padrão para a troca entre diversos tipos de bancos de dados e páginas Web
- ▶ a XML combina modelos para documentos com modelos para bancos de dados
- ▶ hoje, outros formatos para dados semiestruturados na Web são bastante usados, como o JSON.

# Breve história das aplicações de banco de dados

## Novas aplicações (anos 2000 em diante)

- ▶ Aplicações científicas
- ▶ Armazenamento e recuperação de imagens e vídeos
- ▶ Aplicações de *mineração de dados*
- ▶ Aplicações *espaciais* (com localização espacial dos dados)
- ▶ Aplicações de *séries temporais*
- ▶ Aplicações de *Big Data*, sistemas *NoSQL* (= não relacionais)

# Analisando a viabilidade do uso de um SGBD

Os custos envolvidos no uso de um SGBD se relacionam a:

1. Investimentos iniciais em hardware, software e treinamento [dinheiro, tempo]
2. A generalidade que o SGBD fornece para a definição e o processamento de dados [tempo]
3. O esforço adicional necessário para prover segurança, controle de concorrência, recuperação e integridade dos dados [tempo, desempenho]



## Quando é melhor **não** usar um SGBD [convencional]

O uso direto de arquivos ou de SGBDs “não-convencionais” é mais aconselhado que o uso de SGBDs tradicionais (relacionais) nas seguintes situações:

1. O BD e suas aplicações são simples, bem definidos e sem previsão de mudanças
2. A sobrecarga do SGBD pode impedir que requisitos de desempenho (como em programas de tempo-real) sejam atendidos
3. O acesso de múltiplos usuários aos dados não é necessário
4. O dados não “cabem” numa estrutura relacional

# SGBDs mais populares em Fev/2016

295 systems in ranking, February 2016

Rank			DBMS	Database Model	Score		
Feb 2016	Jan 2016	Feb 2015			Feb 2016	Jan 2016	Feb 2015
1.	1.	1.	Oracle	Relational DBMS	1476.14	-19.94	+36.42
2.	2.	2.	MySQL +	Relational DBMS	1321.13	+21.87	+48.67
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1150.23	+6.16	-27.26
4.	4.	4.	MongoDB +	Document store	305.60	-0.43	+38.36
5.	5.	5.	PostgreSQL	Relational DBMS	288.66	+6.26	+26.32
6.	6.	6.	DB2	Relational DBMS	194.48	-1.89	-7.94
7.	7.	7.	Microsoft Access	Relational DBMS	133.08	-0.96	-7.47
8.	8.	8.	Cassandra +	Wide column store	131.76	+0.81	+24.68
9.	9.	9.	SQLite	Relational DBMS	106.78	+3.04	+7.22
10.	10.	10.	Redis +	Key-value store	102.07	+0.92	+2.86
11.	11.	11.	SAP Adaptive Server	Relational DBMS	80.03	-3.15	-6.30
12.	12.	↑ 16.	Elasticsearch +	Search engine	77.84	+0.63	+25.01
13.	↑ 14.	13.	Teradata	Relational DBMS	73.38	-1.57	+3.93
14.	↓ 13.	↓ 12.	Solr	Search engine	72.27	-3.12	-9.21
15.	15.	↑ 17.	Hive	Relational DBMS	52.77	-0.81	+16.21
16.	16.	↓ 14.	HBase	Wide column store	52.02	-1.34	-5.12

<http://db-engines.com/en/ranking>

## Referências Bibliográficas

- ▶ *Sistemas de Bancos de Dados* (6ª edição), Elmasri e Navathe. Pearson, 2010.  
Capítulo 1
- ▶ *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...

- ▶ Conceitos e arquitetura do sistema de banco de dados