

# Introdução aos Sistemas de Bancos de Dados

# Introdução aos Sistemas de Bancos de Dados

- Dados é um tema antigo, porém bastante atual.
- A ascensão da Inteligência Artificial e dos métodos de análise de dados, mesmo que com resultados bastante discutíveis, é assunto central tanto no mundo acadêmico como no meio empresarial e nos governos do mundo todo.

# Introdução aos Sistemas de Bancos de Dados

- Livros como **Moneyball: The Art of Winning an Unfair Game**<sup>1</sup> tornaram-se as bíblias modernas.
- Um artigo de Allen Barra para o The Guardian<sup>2</sup> faz uma crítica bastante severa ao livro e polemiza se a abordagem baseada em estatísticas apresentada realmente condiz com a realidade.
- A própria realidade mostrou que alguns pontos do livro não são verdades absolutas.
- O método Moneyball se opõe a *sacrifice bunts* e *sacrifice flies*<sup>3</sup>, mas o Arizona Diamondbacks chegou a World Series, coisa que o inventor do método nunca conseguiu, usando *bunts*.

---

<sup>1</sup>Adaptado para o cinema.

<sup>2</sup>Moneyball: was the book that changed baseball built on a false premise? em 21 de abril de 2017.

<sup>3</sup>Jogadas do beisebol.

# Introdução aos Sistemas de Bancos de Dados

- Dizer que está tomando decisões baseadas em evidências é um grande negócio.
- Na prática, a tomada de decisões utilizando dados é muito mais complexa, cara e difícil que o apresentado.
- Há também o agravante que, mesmo que o modelo adotado seja correto, o resultado não seja o esperado, pois sempre há riscos e, dada a complexidade do mundo real, é impossível mapear todas as variáveis possíveis.
- Um termo bem sugestivo na estatística, e que mostra como a incerteza ocorre, é **esperança**.

# Introdução aos Sistemas de Bancos de Dados

- A análise de dados possui dois problemas de grande complexidade.
- O primeiro deles é estabelecer um modelo matemático correto.
- Livros de ciências dos dados colocam como requisito não apenas o conhecimento em computação ou engenharia, mas fundamentos sólidos em matemática e estatística.

# Introdução aos Sistemas de Bancos de Dados

- O segundo problema é os sistemas de bancos de dados.
- Modelos estatísticos necessitam de volumes massivos de dados.
- Coletar e organizar estes dados é um grande desafio e não deve ser tratado de maneira simplória.
- Até mesmo sistemas pequenos podem carregar imensa complexidade.

# Introdução aos Sistemas de Bancos de Dados

- Como dito, as bases de dados não são coisas novas. Pelo contrário, já existem a milênios.
- A República Romana já armazenava os dados dos seus cidadãos através do censo. Isto permitia classificá-los de acordo com suas posses.

# Introdução aos Sistemas de Bancos de Dados

Antes de continuar a discussão, porém, é necessário definir o que é uma base de dados:

## Definição

*Uma base de dados é uma coleção de dados relacionados que é projetada, construída e povoada a partir de dados logicamente coerentes entre si, com propósito específico e que representam algum aspecto do mundo.<sup>a</sup>*

---

<sup>a</sup>ELMASRI E NAVATHE, 2011.



# Introdução aos Sistemas de Bancos de Dados

- Apesar da definição, não é raro chamarmos por banco (ou base) de dados os Sistemas de Gerenciamento de Bancos de Dados (SGBDs).
- Um SGBD é um *software* que fornece serviços para "definir, construir, manipular e compartilhar bases de dados com vários usuários e aplicações".<sup>4</sup>

---

<sup>4</sup>ELMASRI E NAVATHE, 2011.

# Introdução aos Sistemas de Bancos de Dados

- Por **definir** tem-se a definição dos metadados.
- Metadados são as especificações de tipos, estruturas e restrições dos dados.
- Por exemplo, tenha-se um sistema acadêmico, a definição de quais dados dos alunos serão armazenados e que o nome do aluno deve ser uma cadeia de caracteres (*string*) de até 300 caracteres são exemplos de metadados.

# Introdução aos Sistemas de Bancos de Dados

- A **construção** da base de dados é o armazenamento de dados segundo sua especificação em um ambiente controlado pelo SGBD.
- A **manipulação** da base de dados corresponde às operações CRUD (*Create*, *Retrieve*, *Update* e *Delete*).
- O **compartilhamento** da base de dados envolve permitir acesso de diversos usuário e aplicações à base concorrentemente.

# Sistemas de Gerenciamento de Bancos de Dados

- Do ponto de vista macro, os SGBDs fornecem uma camada de abstração entre as aplicações e as bases de dados.
- A aplicação não necessita conhecer como os dados estão armazenados em baixo nível, ela apenas solicita os dados aos SGBD.
- O projeto do banco de dados está desacoplado da aplicação, sendo necessário apenas o mapeamento entre os dados organizados no banco e as variáveis ou objetos da aplicação.

# Sistemas de Gerenciamento de Bancos de Dados

- O SGBD possui um catálogo de dados do banco.
- Este catálogo (metadados) é uma estrutura de dados que fornece informações sobre como os dados estão logicamente organizados.
- Como os SGBDs são agnósticos em relação à base de dados, o catálogo de dados é bastante genérico e flexível, permitindo a modelagem de diferentes base de dados, porém ainda submissas ao paradigma adotado pelo SGBD.

# Transações

- Outro conceito importante implementado pelos SGBDs modernos é o de transação.
- Uma transação é um conjunto de ações que envolve um ou mais acessos ao banco de dados, como a leitura ou modificação dos registros.

# ACID

- É desejável que os SGBDs garantam que suas transações atendam às propriedades ACID (*Atomicity, Consistency, Isolation e Durability*).
- Nem todos os SGBDs garantem as propriedades.

# Atomicidade

- Atomicidade é uma propriedade desejável para as transações.
- Uma transação atômica é uma transação que ou é executada por inteiro ou não é executada.
- Em outras palavras, a transação não deve ser executada "pela metade".
- Imagine, por exemplo, uma operação de transferência de dinheiro, onde apenas o ato de retirar de uma conta é executado, mas a inserção do valor em outra não.



# Consistência

- O SGBD também deve buscar manter o base de dados em estado consistente.
- Isto significa que os dados não devem assumir valores que violem as regras de negócio.
- Caso uma transação leve o banco de dados a um estado inconsistente, ela deve ser desfeita.

# Isolamento

- As transações são, principalmente em ambientes concorrentes (onde há múltiplos usuários requisitando e modificando dados da base), preferencialmente isoladas.
- Isto significa que se duas transações desejam modificar um determinado dado, uma deve ser executada apenas após a primeira terminar.
- Caso contrário, se a primeira transação falhar e tiver de ser desfeita, poderá ser impossível retornar ao estado inicial.

# Durabilidade

- Por fim, é esperado que toda transação tenha efeito persistente.
- Isto quer dizer que a aplicação deve esperar que em caso de sucesso, a transação é gravada em disco (ou outro meio de armazenamento persistente).

# Serviços do SGBD

- O SGBD, no entanto é mais do que uma simples camada de abstração.
- Ele é um servidor e, por isso, provê uma série de serviços.

# Controle de redundância

- O primeiro deles é o controle de redundância.
- Tenha-se uma universidade privadas e seus registros.
- Os setores acadêmico e financeiro devem possuir os dados de um mesmo estudante.
- Estes dados tendem a ser replicados (redundantes).

# Controle de redundância

- Sem um SGBD, qualquer modificação em um dado redundante levará a duplo esforço e abre a possibilidade para inconsistência de dados, ou seja, dados que deveriam ser idênticos, mas não são.
- Imagine que um aluno desta universidade se matricule em mais disciplinas.
- Se estes dados, alterados no setor acadêmico, não forem também modificados no setor financeiro, haverá inconsistência dos dados e o valor da mensalidade poderá não ser o correto.

# Controle de redundância

- Os SGBDs, especialmente os relacionais, permitem o projeto de bancos de dados normalizados, ou seja, sem redundâncias.
- Entretanto, em alguns casos, será necessário recorrer à desnormalização para melhoria na performance de consultas.
- Isto significa estabelecer uma redundância controlada.
- É esperado de um bom SGBD ferramentas para lidar com esta redundância.

# Armazenamento persistente de objetos

- Alguns SGBDs possuem suporte ao armazenamento de objetos.
- Apesar de ser uma característica comum aos SGBDs orientados a objetos, SGBDs relacionais já implementam técnicas e ferramentas que aproximam o paradigma relacional ao orientados a objeto, sendo conhecidos como SGBDs objeto-relacionais.
- Deste modo, reduz-se o problema de divergência de impedância causado pela diferença entre o paradigma orientado a objetos, presente na maior parte das aplicações, e o paradigma relacional, presente nos SGBDs mais usados.



# Otimização de consultas

- Os SGBDs permitem a otimização de consultas através de diversas técnicas.
- A primeira delas é a indexação através de tabelas *hash* ou árvore.
- Outra técnica é o uso de *buffers* ou *caches* para armazenar registros e índices na memória, reduzindo o *overhead* da busca em disco.
- Por último, os SGBDs também possuem otimizações de baixo nível da linguagem SQL.

# Backup e recuperação de falhas

- Falhas de *hardware* e *software* são inevitáveis.
- É esperado de todo bom SGBD ferramentas para implementação de políticas de *backup* e recuperação de falhas.

# Quando não usar SGBDs

- Haverá casos onde usar um SGBD não é recomendado.
- Estes são casos específicos.
- Principalmente em sistemas embarcados, onde há enormes restrições de espaço e performance, os SGBDs podem não ser bem vindos.
- Como tudo na computação, a decisão dependerá do contexto, mais especificamente dos requisitos não-funcionais.

# Referências

- Elmasri, Ramez; e Navathe, Shamkant. Sistemas de banco de dados. 2011. 6ª ed. Pearson.
- Barra, Allen. Moneyball: was the book that changed baseball built on a false premise?. The Guardian de 21 de abril de 2017.
- Garcia-Molina, Hector; Ullman, Jeffrey D.; e Widom, Jennifer. Database Systems: The Complete Book. 2009. 2ed. Pearson.