



**UNIVERSIDADE FEDERAL DO  
MARANHÃO - UFMA**

**Banco de Dados II**

# **Processamento de Transações**

**Carlos Eduardo Portela Serra de Castro**

\*

# Processamento de Transações

## Sumário

Introdução

Definição

Problemas de consistência

Recuperação

Estados

*System Log*

*Commit Point*

Propriedades

# Introdução

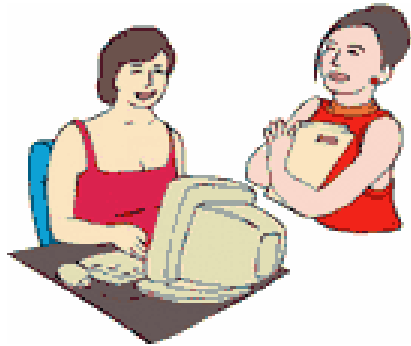
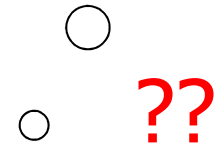
- SGBD
- Número de usuários concorrentes
  - Monousuário x Multiusuário
- Transação

# Porque Banco de Dados?

- Idéia básica:  
Coletar, processar e disseminar dados.
- Necessidade de **armazenar** as informações (*conjunto de dados*) para **tratá-las** e **utilizá-las** de maneira otimizada.

# Onde se encaixa o BD?

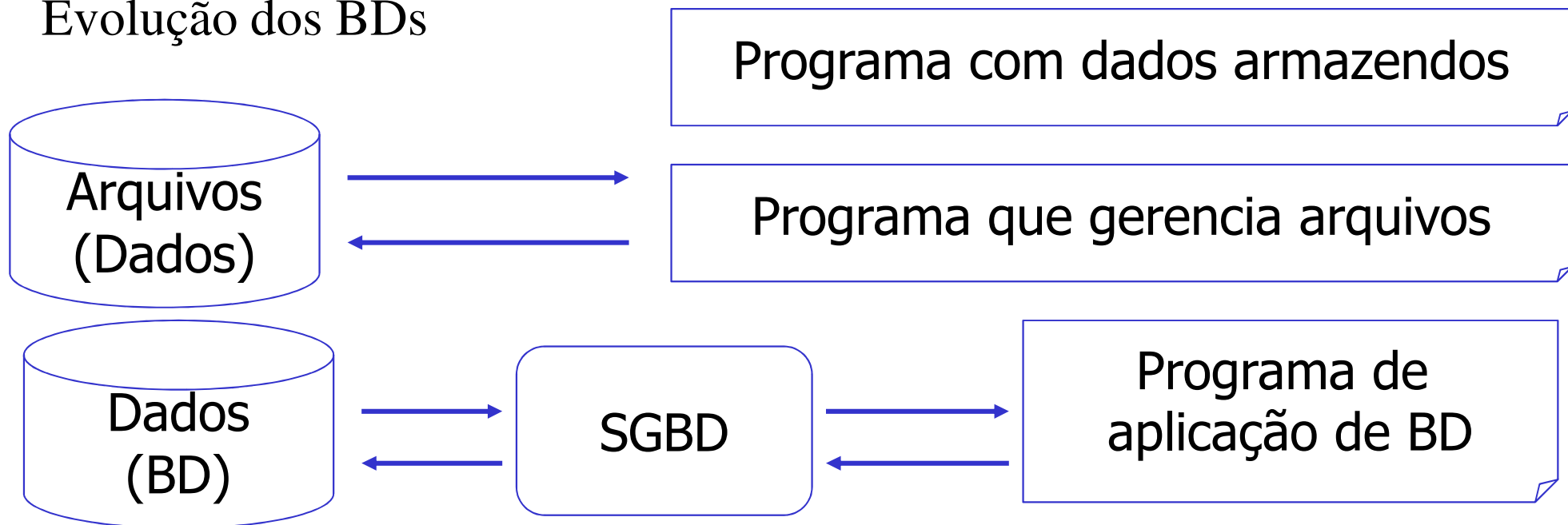
Desenvolver um **Sistema de Informação** consiste em traduzir um **problema** do mundo real em uma **solução** que envolve a organização, pessoas e tecnologia (hardware e software).



Para se implementar um SI, são necessários: Definição das Interfaces, Desenvolvimento das Aplicações, **Projeto do Banco de Dados**.

# O que é Banco de Dados?

- Coleção de dados relacionados.
- Projetado, construído e povoado (com dados) para uma finalidade específica
  - Representa algum aspecto do mundo real, chamado **mini-mundo** ou **universo de discurso**
- Evolução dos BDs



# O que é Banco de Dados?

- *“Uma coleção de dados operacionais inter-relacionados e persistentes. Estes dados são gerenciados de forma independente dos programas que os utilizam, servindo assim a múltiplas aplicações de uma Organização.”*
- Vantagens do uso
  - Dados armazenados em um único local – evita-se redefinições; minimiza-se redundância;
  - Dados compartilhados pelas aplicações – facilita integração de aplicações; evita redefinição de dados;
  - Dados mais independentes das aplicações – novas operações de manipulação de dados não requerem modificação pesada no código da aplicação – aplicações não se preocupam mais com o gerenciamento dos dados;
  - Maior flexibilidade de acesso – linguagens para BD.

# Sistema de Gerência de Banco de Dados (SGBD)

- Coleção de programas que facilitam o processo de **Definir**, **Construir**, **Manipular** um Banco de Dados.
- *“Sistema cujo objetivo principal é gerenciar o **acesso** e a correta **manutenção** dos dados armazenados em um banco de dados.”*
- Funções básicas
  - Métodos de acesso
  - Integridade Semântica
  - Segurança
  - Concorrência
  - Independência – Abstração dos dados

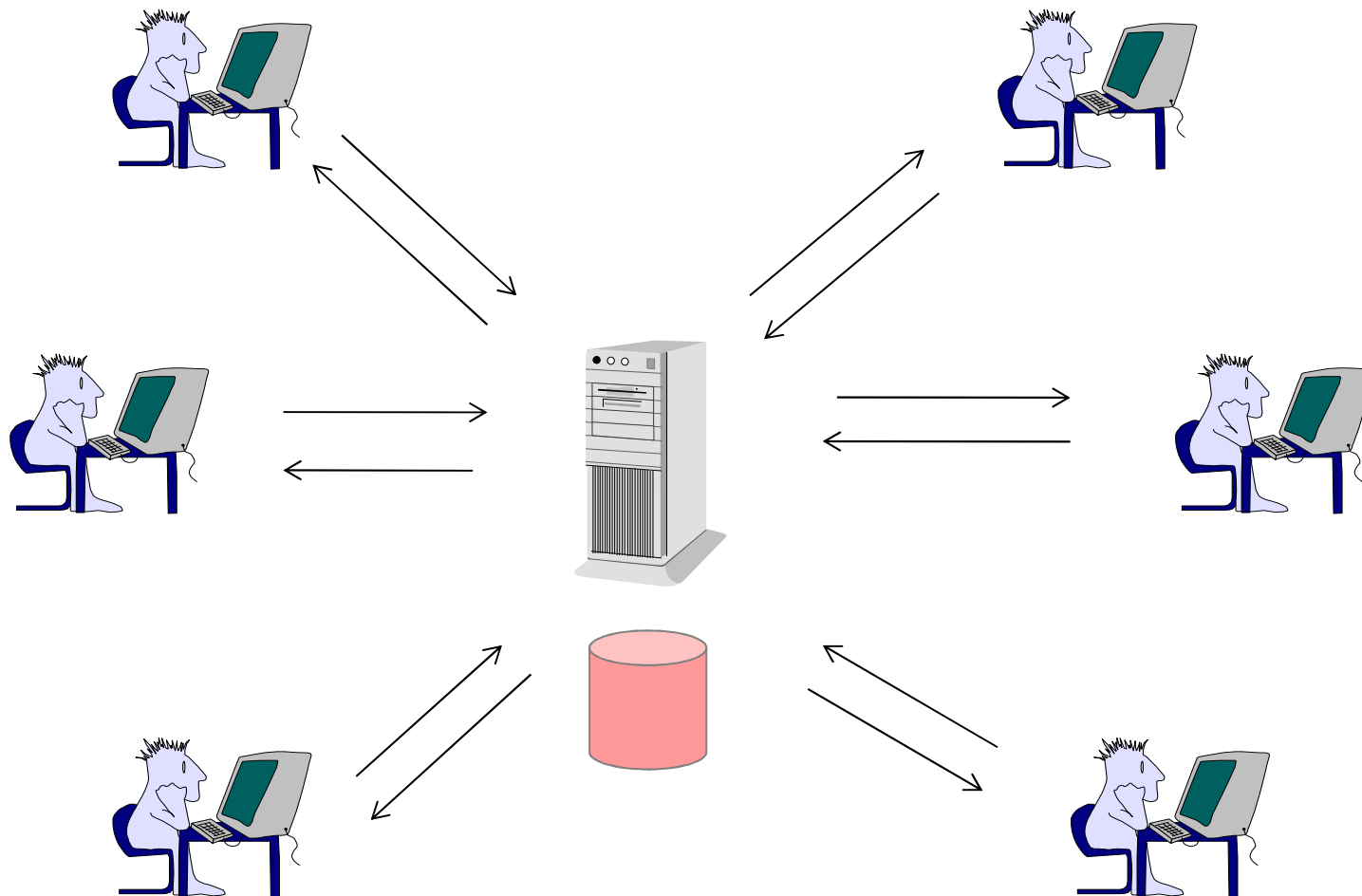


# Sistema de Gerência de Banco de Dados (SGBD)

- Coleção de programas que facilita o processo de **Definição**, **Construção** e **Manipulação** de um Banco de Dados.
- Sistema de Banco de Dados:

Programa de Aplicação + SGBD + BD

# Número de usuários concorrentes



# Transações Concorrentes

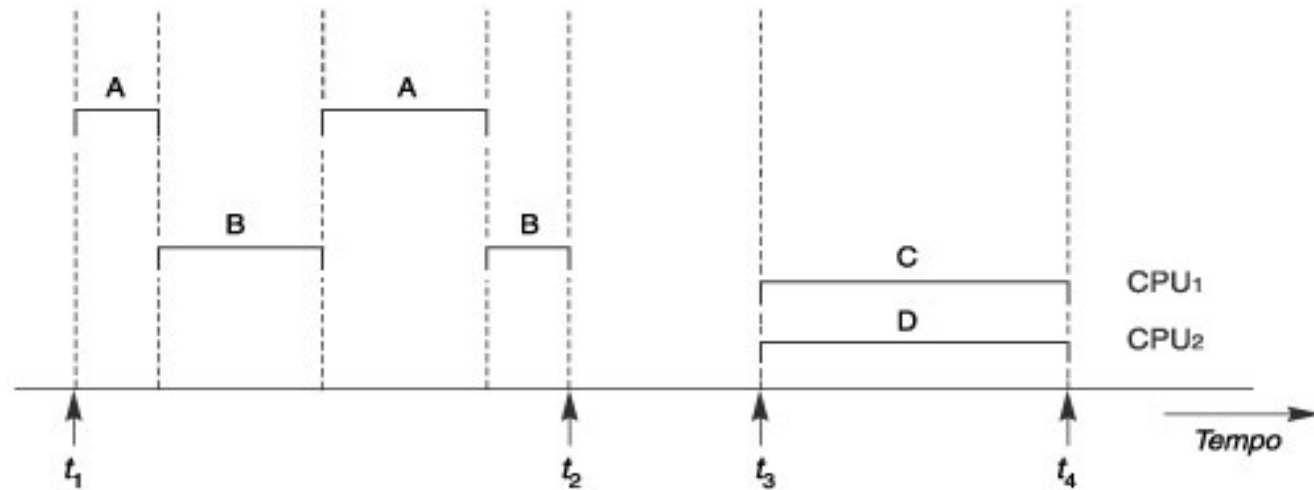


Figura 17.1 Processamento intercalado *versus* processamento paralelo de transações concorrentes.

# Transações

Uma transação é uma unidade lógica do processamento do banco, que inclui uma ou mais operações de acesso ao banco de dados que precisa ser completada (ou desfeita) integralmente para garantir precisão.

- Essas operações podem incluir inclusão, exclusão, modificação ou seleção.
- Iniciar: start transaction (begin).
- Finalizar: commit (work).

# Granularidade de Itens de Dados

- Todas as técnicas de controle de concorrência consideram que o banco de dados é formado a partir de uma série de itens de dados com nomes. Um item de dados pode ser escolhido como sendo entre os seguintes:
  - Um valor de um campo de um registro do banco de dados.
  - Um registro do banco de dados.
  - Um bloco de disco.
  - Um arquivo inteiro.
  - Todo o banco de dados.

# Transações

(a)  $T_1$

---

ler\_item (X);  
X:=X-N;  
escrever\_item (X);  
ler\_item (Y);  
Y:=Y+N;  
escrever\_item (Y);

(b)  $T_2$

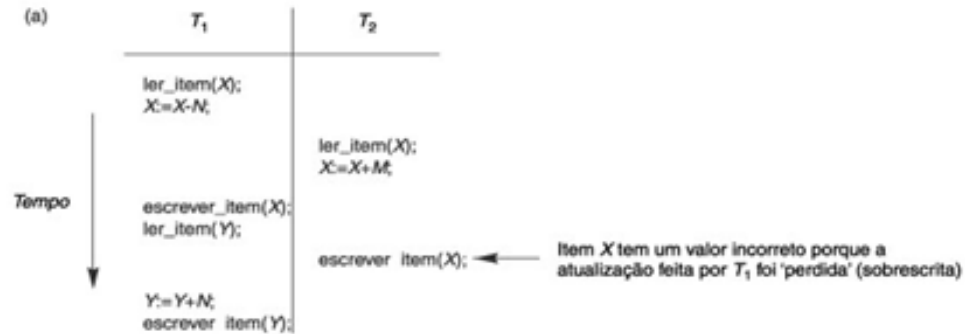
---

escrever\_item (X);  
X:=X+M;  
escrever\_item (X);

Figura 17.2 Duas transações simples. (a) Transação  $T_1$ . (b) Transação  $T_2$ .

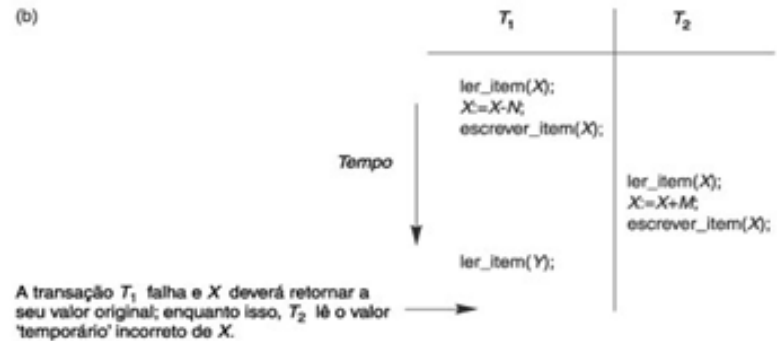
# Problemas de Consistência

- Atualização Perdida: duas transações que ocorrem simultaneamente atualizam o mesmo dado. Isto pode ocorrer em uma seqüência segundo a qual uma das atualizações é perdida.
- Atualização Temporária (Dirty Read): leitura de dados não confirmados de uma linha existente, podendo ocasionar a leitura de uma informação nunca confirmada.
- Leitura Fantasma: na releitura de um conjunto de dados, surgem novas informações no conjunto.
- Leitura Não-Repetida: duas leituras de dados na mesma transação não se repetem. Na segunda leitura, dados não existem ou foram modificados.

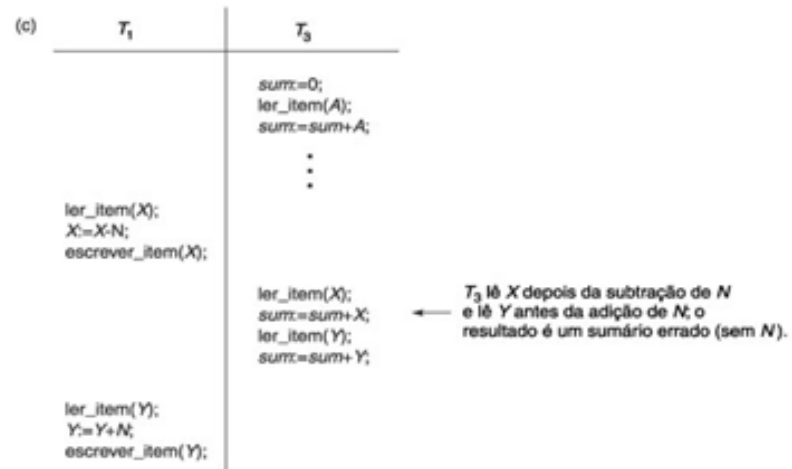


**(a) O problema da atualização perdida.**





## (b) O problema da atualização temporária.



**(c) O problema do sumário incorreto.**

# Estados da Transação

Ativa

Em processo de efetivação

Efetivada

Em processo de aborto

Concluída

## Diagrama de Transição de Estado de uma Transação

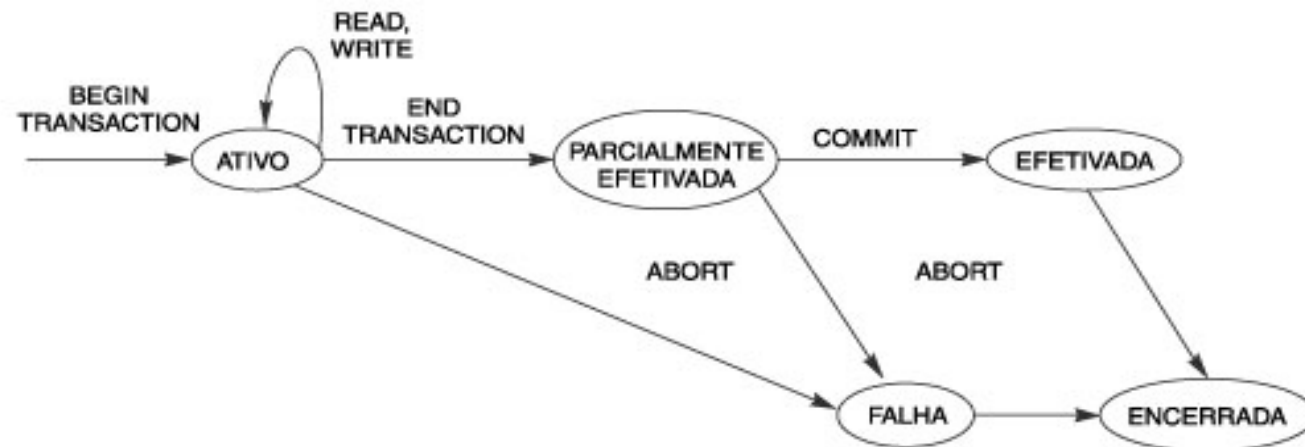


Figura 17.4 Diagrama de transição de estado ilustrando os estados de execução de uma transação.

# *System Log*

- Registro de Ocorrências
- Diário do SGBD
- Log

# Log

- Para auxiliar na recuperação de falhas.
- Log: guarda as informações de todas as operações que afetam o BD.
- Tipos de entradas no log:
  - [start\_transaction, T]
  - [write\_item, T, X, valor antigo, valor novo]
  - [read\_item, T, X]
  - [commit, T]
  - [abort, T]      Obs.: o log pode ser reduzido apenas a operações de gravação

# Log

- Antes da transação alcançar seu ponto de *commit*, qualquer porção do log que não tenha sido escrito no disco deve ser escrito - processo de forçar a escrita.
- *Checkpoints*: outro tipo de entrada no Log
  - todas as transações que tenham as entradas [commit,T] no log antes da entrada checkpoint não precisarão ser refeitas.

# Log

- SGBD deve decidir qual o intervalo entre um *checkpoint* e outro (medido em tempo ou número de transações “*committed*”).
- Ações do checkpoint:
  - Suspender a execução das transações temporariamente;
  - Forçar a escrita de todas as operações de atualização das transações *committed* ;
  - Escrever o registro *checkpoint* no log e forçar a escrita no log;
  - Retomar a execução das transações.



# *Commit Point*

- Transação efetivada.
- Gravação Forçada (force-writting).

# Propriedades ACID

- **Atomicidade:** uma transação é uma unidade de processamento, é realizada integralmente ou não é realizada.
- **Consistência:** uma transação leva um banco de dados de um estado consistente para outro estado consistente.
- **Isolamento:** uma transação deve parecer como se estivesse sendo executada isoladamente.
- **Durabilidade:** as alterações aplicadas a um banco de dados por meio de uma transação confirmada (committed) devem persistir no banco de dados.

# Atomicidade

- Princípio do “*Tudo ou Nada*”
  - ou todas as operações da transação são efetivadas com sucesso no BD ou nenhuma delas se efetiva.
    - preservar a integridade do BD.
- Responsabilidade do subsistema de recuperação contra falhas (*subsistema de recovery*) do SGBD.
  - *desfazer* as ações de transações parcialmente executadas.

# Consistência

- Uma transação sempre conduz o BD de um estado consistente para outro estado também consistente.
- Responsabilidade conjunta do
  - DBA
    - definir todas as RIs para garantir estados e transições de estado válidos para os dados.
      - exemplo: saldo > 0
  - subsistema de *recovery*
    - desfazer as ações da transação que violou a integridade.

# Isolamento

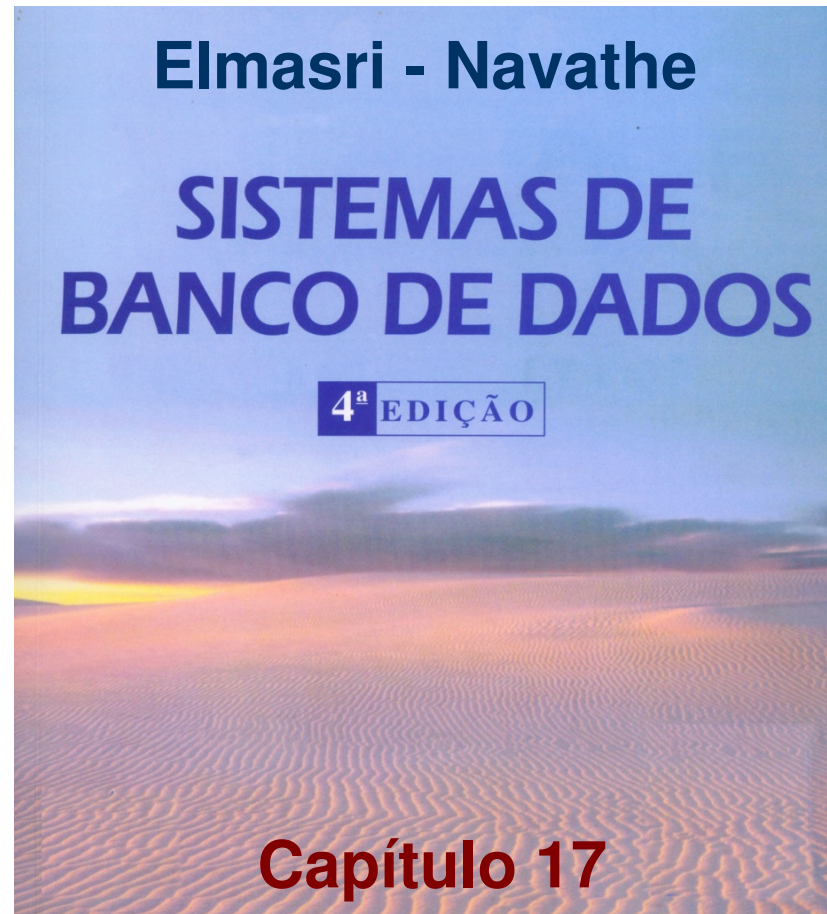
- No contexto de um conjunto de transações concorrentes, a execução de uma transação  $T_x$  **deve funcionar como se  $T_x$  executasse de forma isolada.**
  - $T_x$  **não deve sofrer interferências** de outras transações executando concorrentemente.
- Responsabilidade do subsistema de controle de concorrência (*scheduler*) do SGBD.
  - garantir escalonamentos **sem interferências.**

# Durabilidade

- Deve-se garantir que as **modificações realizadas por uma transação que concluiu com sucesso persistam no BD.**
  - nenhuma falha posterior ocorrida no BD deve perder essas modificações.
- Responsabilidade do **subsistema de *recovery*.**
  - **refazer** transações que executaram com sucesso em caso de falha no BD.

# Níveis de Isolamento

- Nível 0: não sobrescreve leitura de sujeira de transações de nível mais alto;
- Nível 1: não permite atualizações perdidas;
- Nível 2: não há atualizações perdidas nem leitura de sujeira;
- Nível 3: nível 2 + leituras repetíveis.  
(também conhecido como isolamento verdadeiro).



**Introdução aos Conceitos e à  
Teoria do Processamento de Transações**