

# Processamento de Transações

## Banco de Dados: Teoria e Prática

André Santanchè e Luiz Celso Gomes Jr  
Instituto de Computação - UNICAMP  
Setembro 2013

# Exercício 1

- Quais as vantagens e desvantagens de se permitir acesso concorrente ao banco de dados?

# Transação

- Execução concorrente de programas é essencial para a boa performance do SGBD
  - Acesso a disco é frequente mas lento → concorrência melhora aproveitamento da CPU
- Perspectivas sobre os dados:
  - Programa do usuário → pode realizar várias operações com os dados
  - SGBD → se preocupa apenas com leituras e gravações

(Ramakrishnan, 2003b)

# Transação e Concorrência

- Transação: visão abstrata do SGBD sobre um programa do usuário:
  - Uma sequência de leituras e gravações
- Perspectivas sobre a transação:
  - Usuário → sua transação sendo executada individualmente
  - SGBD → concorrência intercalando leituras/gravações de várias transações

# Modelo Simplificado do BD

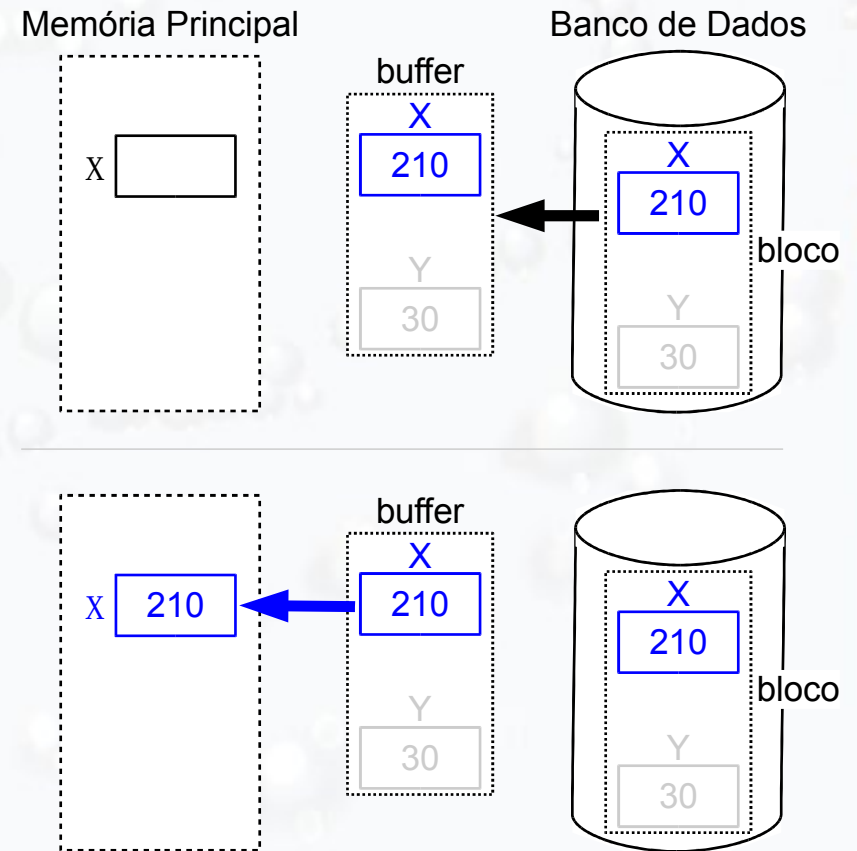
- BD: coleção de itens nomeados
- Conceitos são independentes de granularidade
- Operações:
  - ler(X): lê item X do BD e armazena na variável X do programa
  - gravar(X): grava variável X do programa no item X do BD

(Elmasri, 2010)

# Operação de Leitura Como Acontece

## ■ ler(X)

- encontra bloco X no disco
- copia bloco para buffer da memória principal (se ainda não estiver lá)
- copia o item X do buffer para a variável X da memória principal

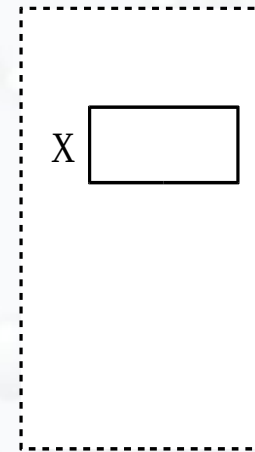


(Elmasri, 2010)

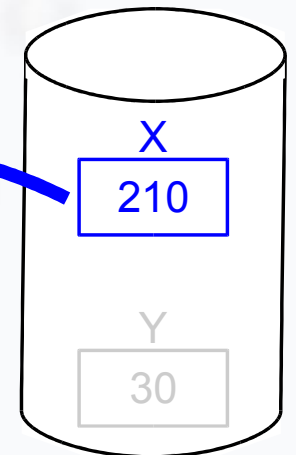
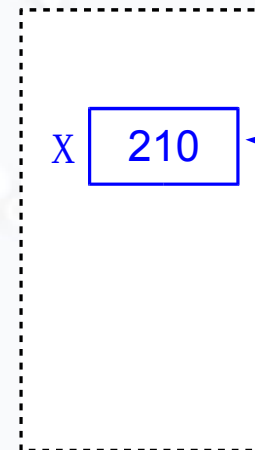
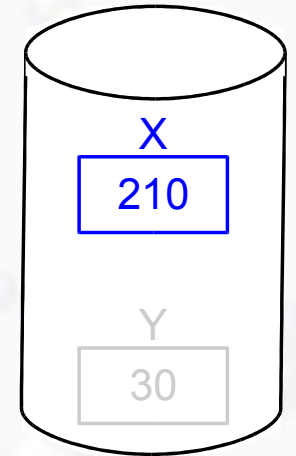
# Operação de Leitura Como Abstraímos

- ler(X)

Memória Principal



Banco de Dados

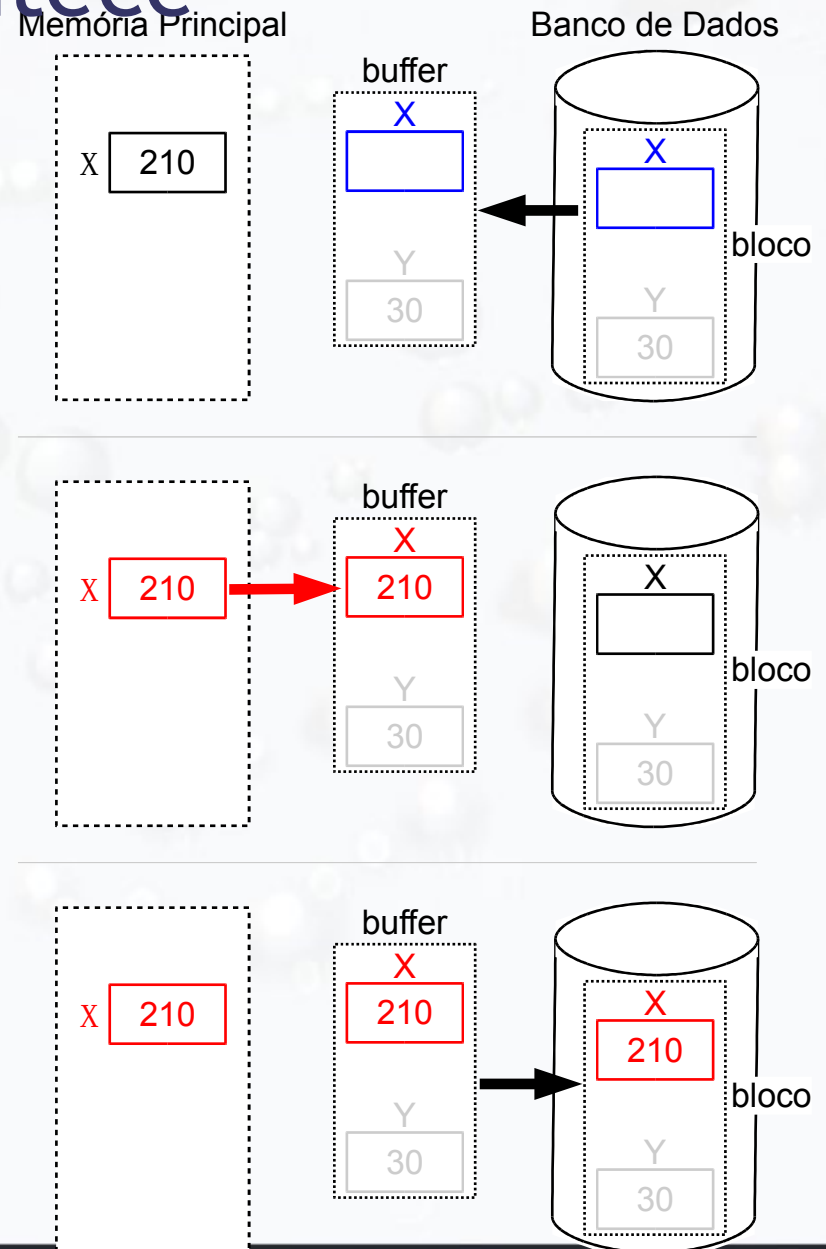


# Operação de Gravação Como Acontece

## ■ gravar(X)

- encontra bloco X no disco
- copia bloco para buffer da memória principal (se ainda não estiver lá)
- copia variável X da memória principal para o buffer
- atualiza o buffer no disco

(Elmasri, 2010)

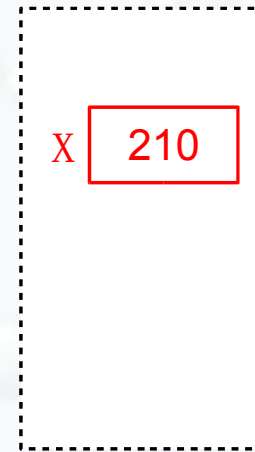




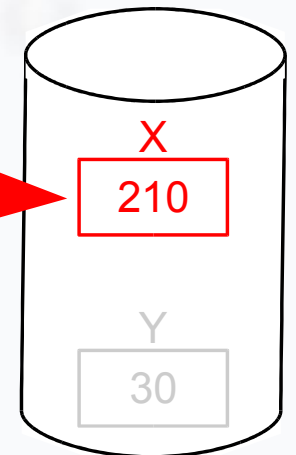
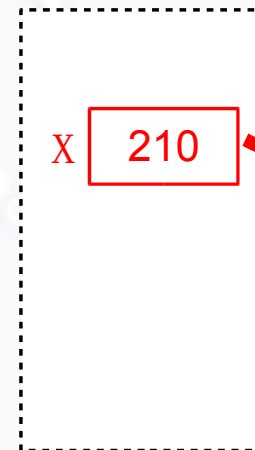
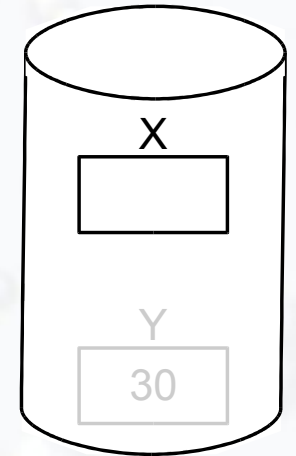
# Operação de Gravação Como Abstraímos

- gravar(X)

Memória Principal



Banco de Dados

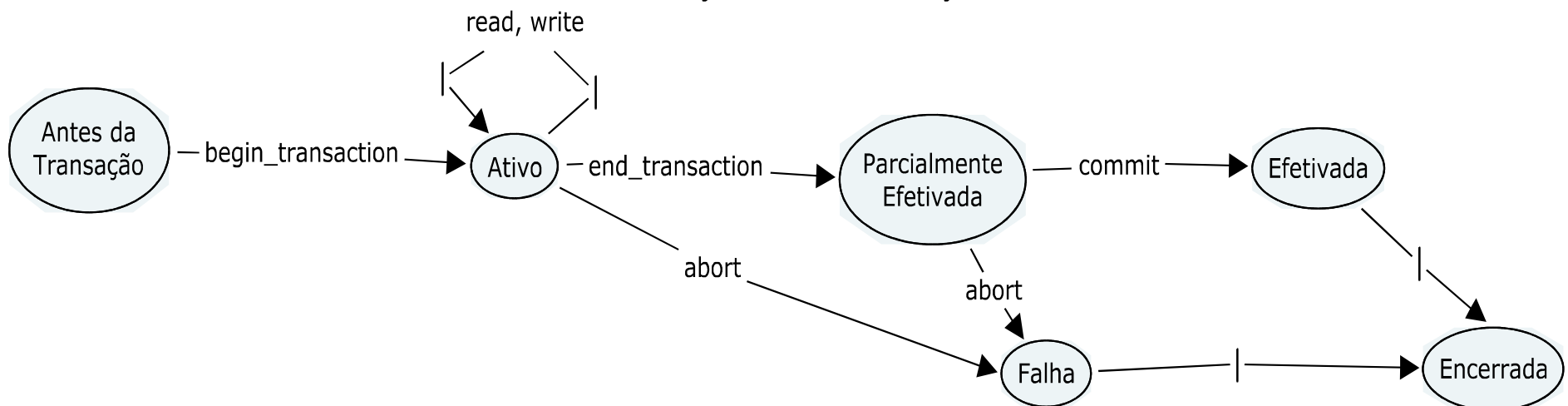


# Transação

## Estados de Execução

- BEGIN\_TRANSACTION
- READ ou WRITE
- END\_TRANSACTION
- COMMIT\_TRANSACTION
- ROLLBACK (ou ABORT)

Transições de uma Transação



# Exemplo

## Transação 1: Transferência

**T1**

—

**ler** (X)

$X = X - N$

**gravar** (X)

**ler** (Y)

$Y = Y + N$

**gravar** (Y)



Prateleira X

transferência



Prateleira Y

# Exemplo

## Transação 2: Aquisição

**T2**  
—  
**ler** (X)  
 $X = X + M$   
**gravar** (X)



aquisição



Prateleira X

# Transações Concorrentes

## Plano de Execução

- Necessidade de um Plano de Execução

T1	T2
<b>ler</b> (X) $X = X - N$	<b>ler</b> (X) $X = X + M$
<b>gravar</b> (X)	<b>gravar</b> (X)
<b>ler</b> (Y) $Y = Y + N$	
<b>gravar</b> (Y)	

# Plano de Execução Serial

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) X = X - N <b>gravar</b> (X) <b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)	<b>ler</b> (X) X = X + M <b>gravar</b> (X)

# Plano de Execução Serial

**S**

---

**ler** (X)

$X = X - N$

**gravar** (X)

**ler** (Y)

$Y = Y + N$

**gravar** (Y)

**ler** (X)

$X = X + M$

**gravar** (X)

# Plano de Execução Intercalado

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) X = X - N	<b>ler</b> (X) X = X + M
<b>gravar</b> (X) <b>ler</b> (Y)	<b>gravar</b> (X)
Y = Y + N <b>gravar</b> (Y)	



# Plano de Execução Intercalado

**S**

---

**ler** (X)

$X = X - N$

**ler** (X)

$X = X + M$

**gravar** (X)

**ler** (Y)

**gravar** (X)

$Y = Y + N$

**gravar** (Y)

# Plano de Execução (*Schedule*)

- Aplicável a várias transações simultâneas
- Lista de ações de conjunto de transações
  - leitura, gravação, abort, commit
- Na *schedule*:
  - $S$  para transações  $T_1, T_2, \dots, T_n$
  - ordem de ações no plano  $T_i =$  ordem das ações em  $S$

# Exemplo

## Transações Concorrentes

- Problemas?

T1	T2
<b>ler</b> (X) $X = X - N$	<b>ler</b> (X) $X = X + M$
<b>gravar</b> (X)	<b>gravar</b> (X)
<b>ler</b> (Y) $Y = Y + N$	
<b>gravar</b> (Y)	

# Problema?

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) $X = X - N$  <b>gravar</b> (X) <b>ler</b> (Y)  $Y = Y + N$ <b>gravar</b> (Y)	<b>ler</b> (X) $X = X + M$  <b>gravar</b> (X)

# Problema

- Atualização  
Perdida

(Elmasri, 2010)

**T1**

**ler** (X)

$X = X - N$

**gravar** (X)

**ler** (Y)

$Y = Y + N$

**gravar** (Y)

**T2**

**ler** (X)

$X = X + M$

**gravar** (X)

# Problema

T1	T2
<b>ler</b> (X) $X = X - N$	
<b>gravar</b> (X) <b>ler</b> (Y)	<b>ler</b> (X) $X = X + M$
$Y = Y + N$ <b>gravar</b> (Y)	<b>gravar</b> (X)

- Sobrescrita de dados alterados sem commit
    - conflito WW (write/write)
- (Ramakrishnan, 2003)

# Problema?

T1	T2
<pre>ler (X) X = X - N gravar (X)</pre>	<pre>ler (X) X = X + M gravar (X)</pre>
<pre>ler (Y) ***crash***</pre>	

# Problema *Dirty Read*

T1	T2
<b>ler</b> (X) X = X - N <b>gravar</b> (X)	<b>ler</b> (X) X = X + M <b>gravar</b> (X)
<b>ler</b> (Y) <b>***crash***</b>	

- Leitura de dados alterados sem commit
  - conflito WR (write/read)(Ramakrishnan, 2003)

- Atualização Temporária

(Elmasri, 2010)



# Exemplo

## Transação 3: Sumário

**T3**

—

soma = 0

**ler** (A)

soma = soma + A

...

**ler** (X)

soma = soma + X

**ler** (Y)

soma = soma + Y

...



Problema?

# Intercalação com Transferência

## Problema?

T1	T3
<pre>ler (X) X = X - N gravar (X)  ler (Y) Y = Y + N gravar (Y)</pre>	<pre>soma = 0 ler (A) soma = soma + A  ...  ler (X) soma = soma + X ler (Y) soma = soma + Y  ...</pre>

# Problema

T1	T3
<b>ler</b> (X) X = X - N <b>gravar</b> (X)	soma = 0 <b>ler</b> (A) soma = soma + A  ...
<b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)	<b>ler</b> (X) soma = soma + X <b>ler</b> (Y) soma = soma + Y  ...

- **Resumo  
Incorreto**  
(Elmasri, 2010)

# Problema

T1	T3
<b>ler</b> (X) X = X - N <b>gravar</b> (X)	soma = 0 <b>ler</b> (A) soma = soma + A  ...  <b>ler</b> (X) soma = soma + X <b>ler</b> (Y) soma = soma + Y  ...
<b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)	

- Leitura de dados alterados sem commit
  - conflito WR

# Exemplo

## Transação 4: Reserva de Livro

**T4**

---

**ler** (B)

verifica (B)

...

**ler** (B)

reserva (B)

**gravar** (B)

# Intercalação da Reserva Problema?

<b>T4</b>	<b>T4'</b>
<b>ler</b> (B) verifica (B)  ...	<b>ler</b> (B) verifica (B)  ...
<b>ler</b> (B) reserva (B) <b>gravar</b> (B)	<b>ler</b> (B) reserva (B) <b>gravar</b> (B)

# Problema

## Leitura Não Repetitiva

T4	T4'
<b>ler</b> (B) verifica (B)	<b>ler</b> (B) verifica (B)
...	...
<b>ler</b> (B) reserva (B) <b>gravar</b> (B)	<b>ler</b> (B) reserva (B) <b>gravar</b> (B)

- Leitura Não Repetitiva
  - conflito RW (read/write)(Ramakrishnan, 2003)
- Leitura Não Repetitiva  
(Elmasri, 2010)

# Problemas com Transações Concorrentes

- Atualização Perdida
- Atualização Temporária
- Resumo Incorreto
- Leitura não repetitiva

(Elmasri, 2010)



# Problemas com Transações Concorrentes

- **Conflito WR**
  - Leitura de dados alterados sem commit
- **Conflito RW**
  - Leitura não repetível
- **Conflito WW**
  - Sobrescrita de dados alterados sem commit

(Ramakrishnan, 2003)

# Propriedades ACID

- **Atomicidade:** todas as operações da transação acontecem ou nenhuma acontece
- **Preservação de Consistência:** a execução completa de uma transação faz o BD passar de um estado consistente para outro
- **Isolamento:** uma transação deve ser executada como se estivesse isolada das demais
- **Durabilidade ou permanência:** se uma transação é efetivada, seu efeito persiste

# Plano de Execução Restaurável

- Plano Restaurável
  - T realiza commit somente depois que todas as transações cujos valores T leu realizam commit
- Plano Livre de Cascata (cascadeless)
  - T só lê valores que foram alterados por transações que já realizaram commit
- Plano Estrito
  - T só lê e/ou grava valores que foram alterados por transações que já realizaram commit

# Plano Serial e Serializável

- Plano Serial
  - Transações completas são executadas em série
  - Não há intercalação de operações entre transações
- Plano Serializável
  - equivalente a algum plano serial

# Plano Serial 1

T1	T2
<p><b>ler</b> (X) X = X - N <b>gravar</b> (X) <b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)</p>	<p><b>ler</b> (X) X = X + M <b>gravar</b> (X)</p>

# Plano Serial 2

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) X = X - N <b>gravar</b> (X) <b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)	<b>ler</b> (X) X = X + M <b>gravar</b> (X)

# Plano Serializável?

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) $X = X - N$  <b>gravar</b> (X) <b>ler</b> (Y)  $Y = Y + N$ <b>gravar</b> (Y)	<b>ler</b> (X) $X = X + M$  <b>gravar</b> (X)

# Não Serializável

T1	T2
<p data-bbox="527 421 932 560"><b>ler</b> (X) <math>X = X - N</math></p> <p data-bbox="527 932 917 1070"><b>gravar</b> (X) <b>ler</b> (Y)</p> <p data-bbox="527 1342 932 1481"><math>Y = Y + N</math> <b>gravar</b> (Y)</p>	<p data-bbox="1102 676 1513 815"><b>ler</b> (X) <math>X = X + M</math></p> <p data-bbox="1102 1187 1491 1246"><b>gravar</b> (X)</p>



# Plano Serializável?

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) $X = X - N$ <b>gravar</b> (X)	<b>ler</b> (X) $X = X + M$ <b>gravar</b> (X)
<b>ler</b> (Y) $Y = Y + N$ <b>gravar</b> (Y)	

# Serializável

<b>T1</b>	<b>T2</b>
<b>ler</b> (X) $X = X - N$ <b>gravar</b> (X)	<b>ler</b> (X) $X = X + M$ <b>gravar</b> (X)
<b>ler</b> (Y) $Y = Y + N$ <b>gravar</b> (Y)	

Plano Serializável  
Grafo de Precedência

# Grafo de Precedência

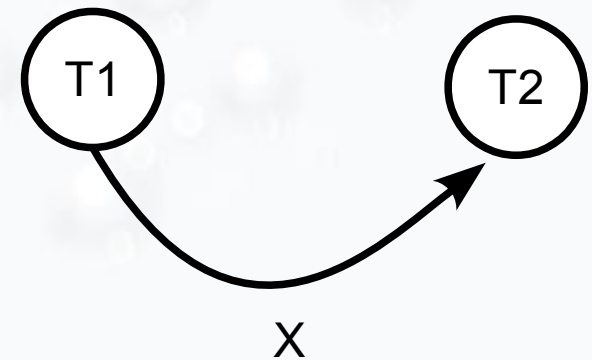
## Algoritmo

- Para cada transação crie um nó no grafo
- Para cada caso em  $S$ 
  - $T_j \rightarrow \text{ler}(x)$  depois de  $T_i \rightarrow \text{gravar}(x)$ 
    - $\text{aresta}(T_i \rightarrow T_j)$
  - $T_j \rightarrow \text{gravar}(x)$  depois de  $T_i \rightarrow \text{ler}(x)$ 
    - $\text{aresta}(T_i \rightarrow T_j)$
  - $T_j \rightarrow \text{gravar}(x)$  depois de  $T_i \rightarrow \text{gravar}(x)$ 
    - $\text{aresta}(T_i \rightarrow T_j)$
- Serializável  $\rightarrow$  sem ciclos

(Elmasri, 2010)

# Plano Serial 1

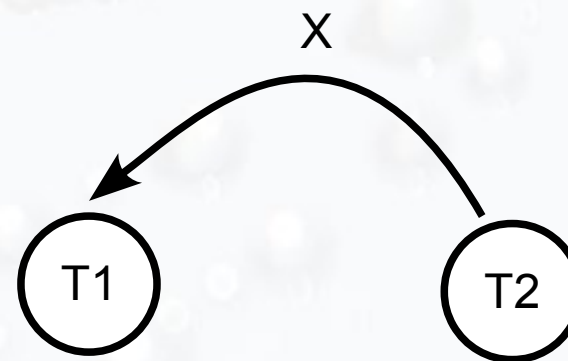
T1	T2
<b>ler</b> (X) X = X - N <b>gravar</b> (X) <b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)	<b>ler</b> (X) X = X + M <b>gravar</b> (X)



Plano Serial 1

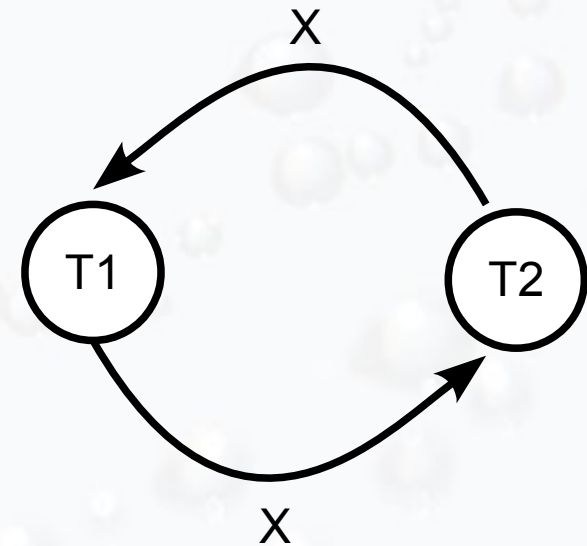
# Plano Serial 2

T1	T2
<p><b>ler</b> (X) X = X - N <b>gravar</b> (X) <b>ler</b> (Y) Y = Y + N <b>gravar</b> (Y)</p>	<p><b>ler</b> (X) X = X + M <b>gravar</b> (X)</p>



# Não Serializável

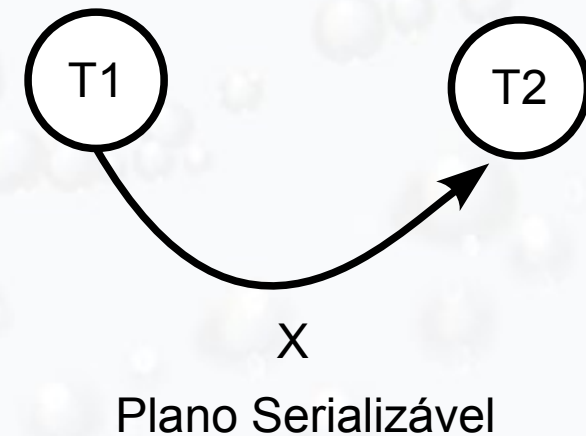
T1	T2
<b>ler</b> (X) $X = X - N$	
<b>gravar</b> (X) <b>ler</b> (Y)	<b>ler</b> (X) $X = X + M$
$Y = Y + N$ <b>gravar</b> (Y)	<b>gravar</b> (X)



Plano Não Serializável

# Serializável

T1	T2
<b>ler</b> (X) $X = X - N$ <b>gravar</b> (X)	
<b>ler</b> (Y) $Y = Y + N$ <b>gravar</b> (Y)	<b>ler</b> (X) $X = X + M$ <b>gravar</b> (X)





## Exercício 2

- Defina se os planos a seguir são seriais ou serializáveis. Desenhe os grafos de precedência.

Obs.:  $r1(x) ==$  Transação 1 lê  $x$ .

a)  $r1(x)$ ,  $w1(y)$ ,  $r3(x)$ ,  $w2(y)$ ,  $w2(y)$

b)  $r1(x)$ ,  $r2(y)$ ,  $w2(y)$ ,  $w1(y)$ ,  $w3(x)$ ,  $r2(x)$

# Equivalência

- Planos Conflito Equivalentes
- Equivalência de Visão

# Plano Conflito Serializável

- Planos Conflito Equivalentes
  - Ordem de operações conflitantes for a mesma em ambos
  - Operações conflitantes
    - pertencem a diferentes transações
    - acessam o mesmo item
    - pelo menos uma for `gravar`
- Plano Conflito Serializável
  - Conflito equivalente a um plano serial

# Grafo de Precedência

## Algoritmo

- Para cada transação crie um nó no grafo
- Para cada caso em  $S$ 
  - $T_i$  precede e conflita com  $T_j$ 
    - aresta( $T_i \rightarrow T_j$ )
- Serializável  $\rightarrow$  sem ciclos

(Ramakrishnan, 2003)

# Equivalência de Visão

- Dois planos  $S$  e  $S'$  possuem equivalência se:
  - Possuem as mesmas transações e operações
  - No plano  $S$ , se há um  $\text{read}(X)$  em  $T_i$  que seja valor original (antes de  $S$ ) ou gravado por um  $\text{write}(X)$  em  $T_j$ , o mesmo acontece em  $S'$
  - No plano  $S$ , se  $\text{write}(Y)$  é a última operação em  $Y$  a gravar em  $T_k$ , o mesmo acontece em  $S'$

# Falha

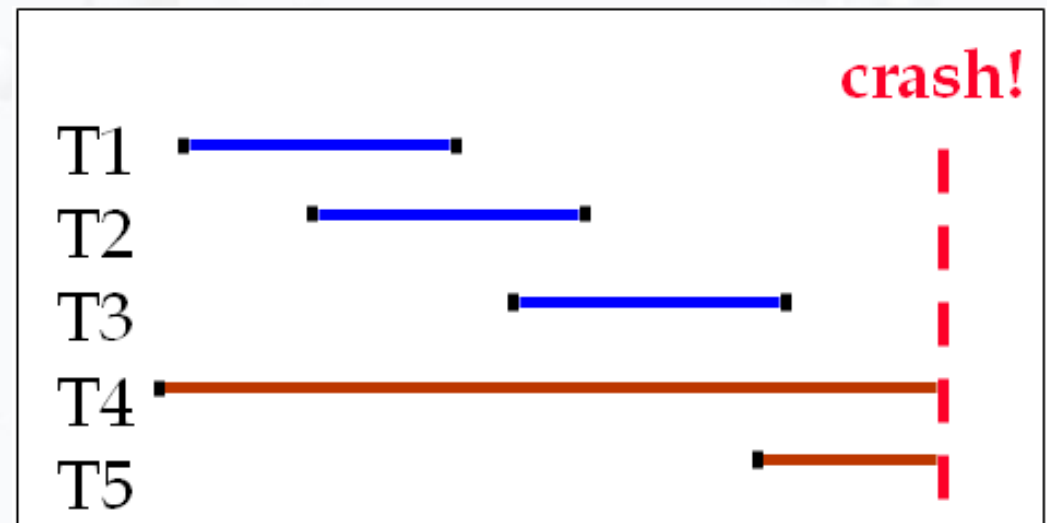
- Tipos de Falha:
  - Sem dano físico ao BD:
    - O computador falhar (crash ou queda de sistema)
    - Um erro de transação ou sistema
    - Erros locais ou condições de exceção detectadas pela transação
    - Imposição do controle de concorrência
  - Com dano físico ao BD:
    - Falha de disco
    - Problemas físicos e catástrofes

# Transação e Atomicidade

- A transação é uma unidade de trabalho atômica:
  - ou é executada completamente ou é não é executada por inteiro
  - transações podem reverter (rollback)

- Exemplos:

- T1, T2 & T3 completas
- T4 & T5 devem ser revertidas



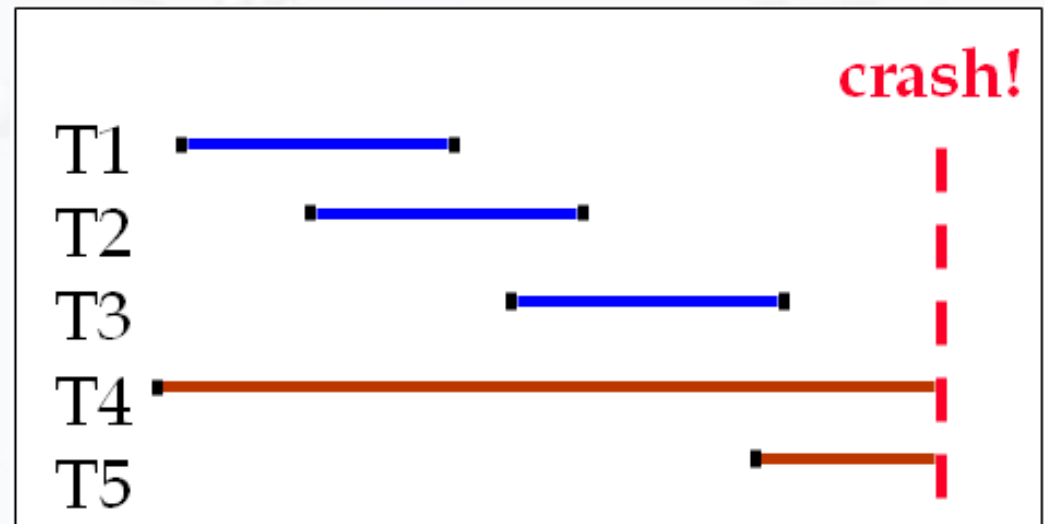
(Ramakrishnan, 2003b)

# Transação e Durabilidade

- O que fazer se o SGBD parar?

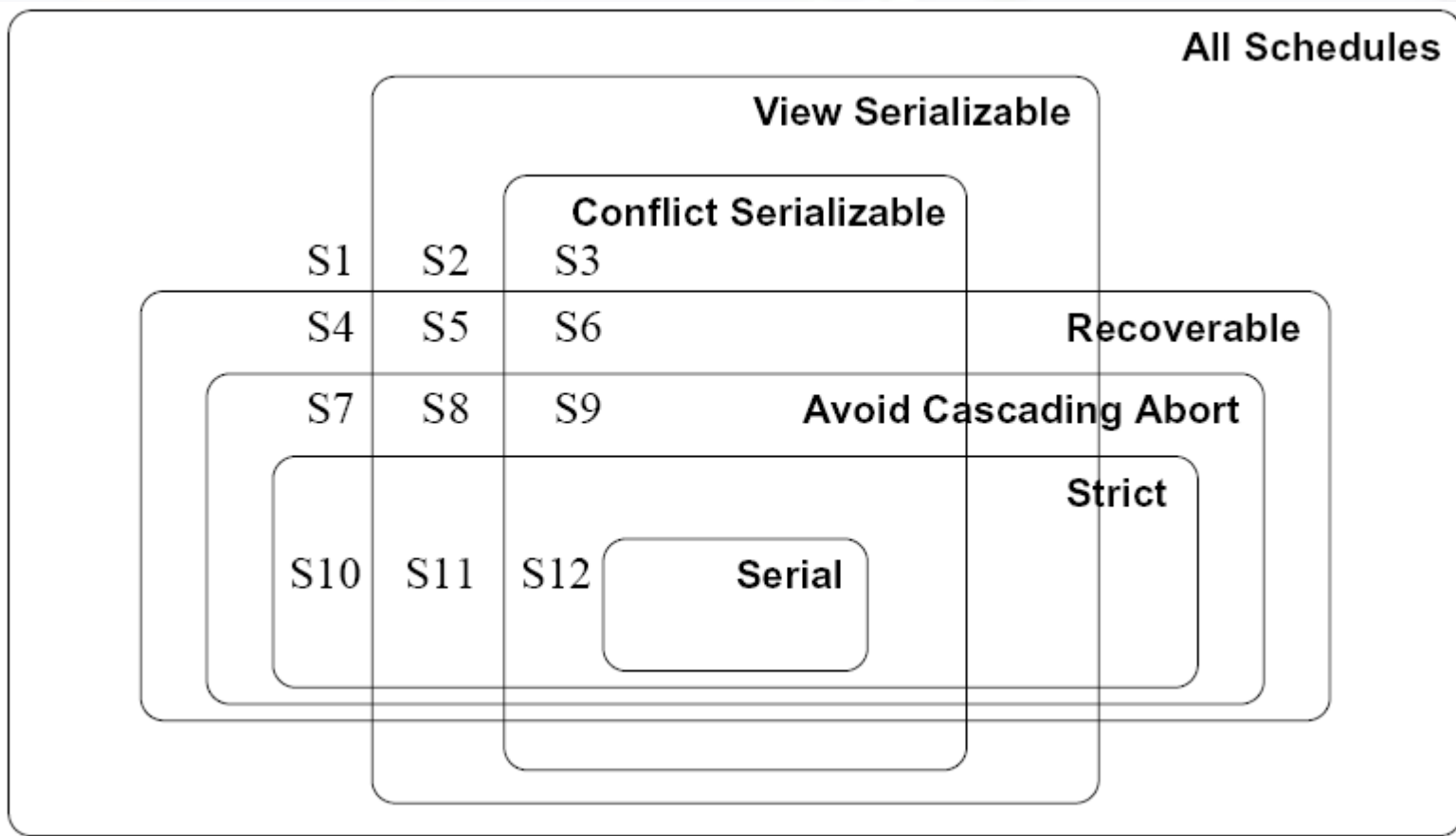
- Exemplos:

- T1, T2 & T3 tem que permanecer



(Ramakrishnan, 2003b)





# Exercício 3

- Para cada propriedade ACID (atomicidade, consistência, isolamento, durabilidade), descreva um problema que pode acontecer caso o SGBD não a garanta.

**André Santanchè**

<http://www.ic.unicamp.br/~santanche>

# Referências

- Elmasri, Ramez; Navathe, Shamkant B. (2005) **Sistemas de Bancos de Dados**. Addison-Wesley, 4<sup>a</sup> edição em português.
- Elmasri, Ramez; Navathe, Shamkant B. (2010) **Sistemas de Banco de Dados**. Pearson, 6<sup>a</sup> edição em português.
- Ramakrishnan, Raghu; Gehrke, Johannes (2003) **Database Management Systems**. McGraw-Hill, 3<sup>rd</sup> edition.
- Ramakrishnan, Raghu; Gehrke, Johannes (2003b) **Database Management Systems**. McGraw-Hill, 3rd edition (companion slides).

# Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.
- Mais detalhes sobre a referida licença Creative Commons veja no link:  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>
- Agradecimentos: fotografia da capa e fundo por Ben Collins -  
<http://www.flickr.com/photos/graylight/>.  
Ver licença específica em  
<http://www.flickr.com/photos/graylight/261480919/>