

Processamento de Transações

Banco de Dados: Teoria e Prática

André Santanchè

Instituto de Computação - UNICAMP

Setembro 2019

Questão

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

Questão

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

□ Vantagens

- Otimização do acesso
- Melhor aproveitamento do sistema
 - Menos tempo ocioso
- Permite colaboração

□ Desvantagens

- Pode gerar inconsistência
- Pode reduzir desempenho
- Pode ter problemas de bloqueio

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

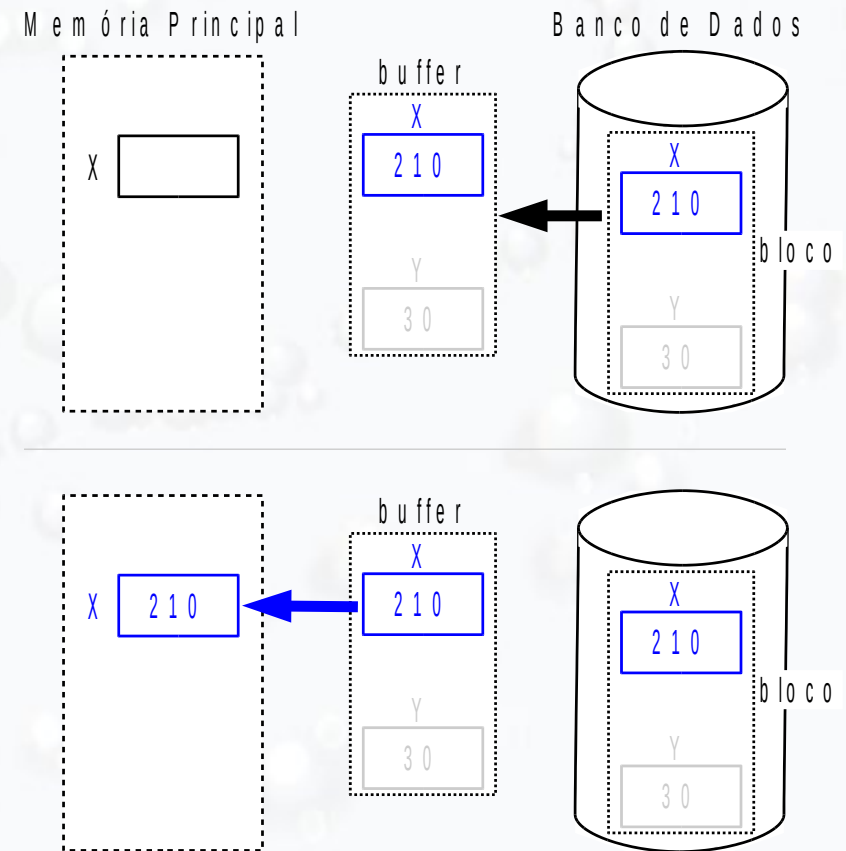
(Elmasri, 2010)

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

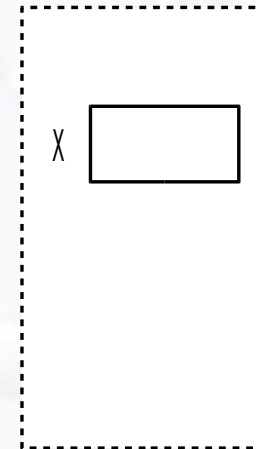
(Elmasri, 2010)



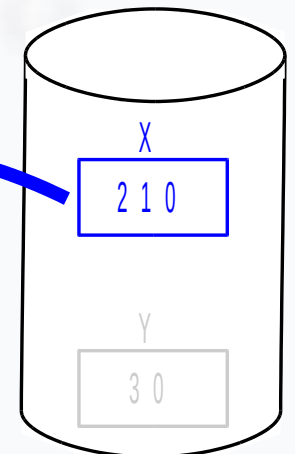
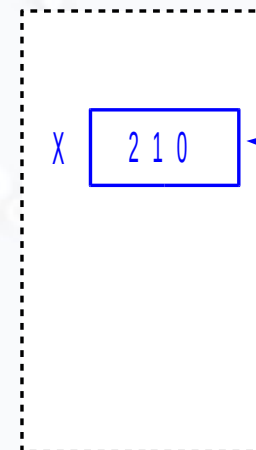
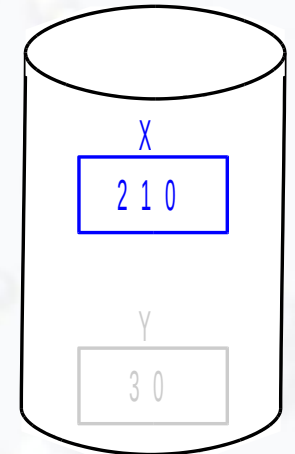
Operação de Leitura Como Abstraímos

■ ler(X)

M em ória P rincipal



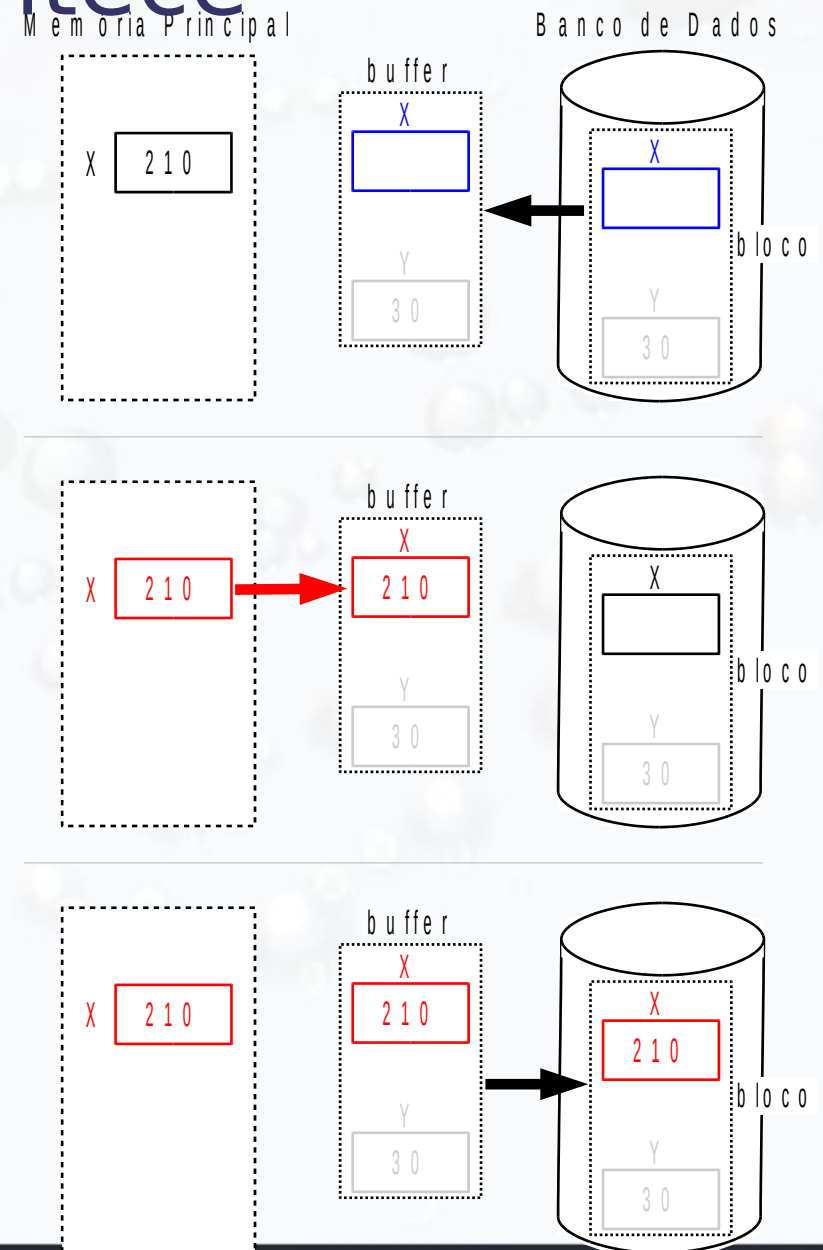
Banco de Dados



Operação de Gravação Como Acontece

■ gravar(X)

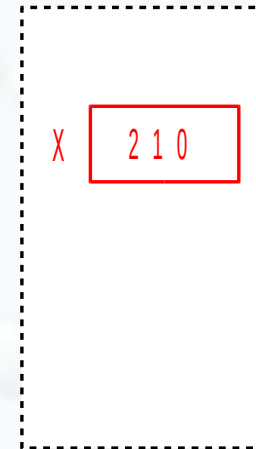
- encontra bloco X no disco
- copia bloco para buffer da memória principal (se ainda não estiver lá)
- copia variável X da memória principal para o buffer
- atualiza o buffer no disco (Elmasri, 2010)



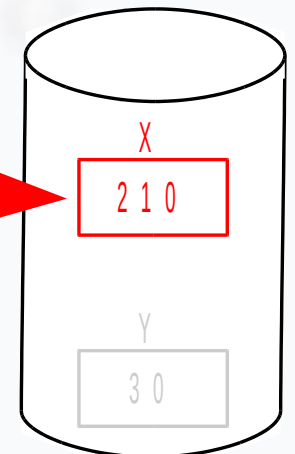
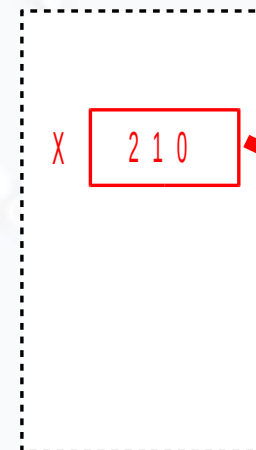
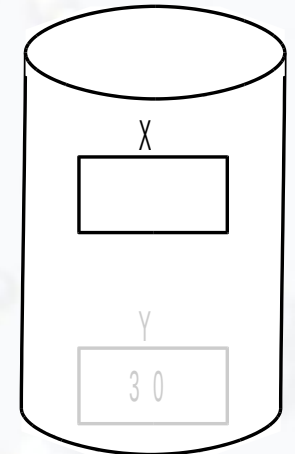
Operação de Gravação Como Abstraímos

■ gravar(X)

Memória Principal



Banco de Dados



Transação

Estados de Execução

■ BEGIN_TRANSACTION

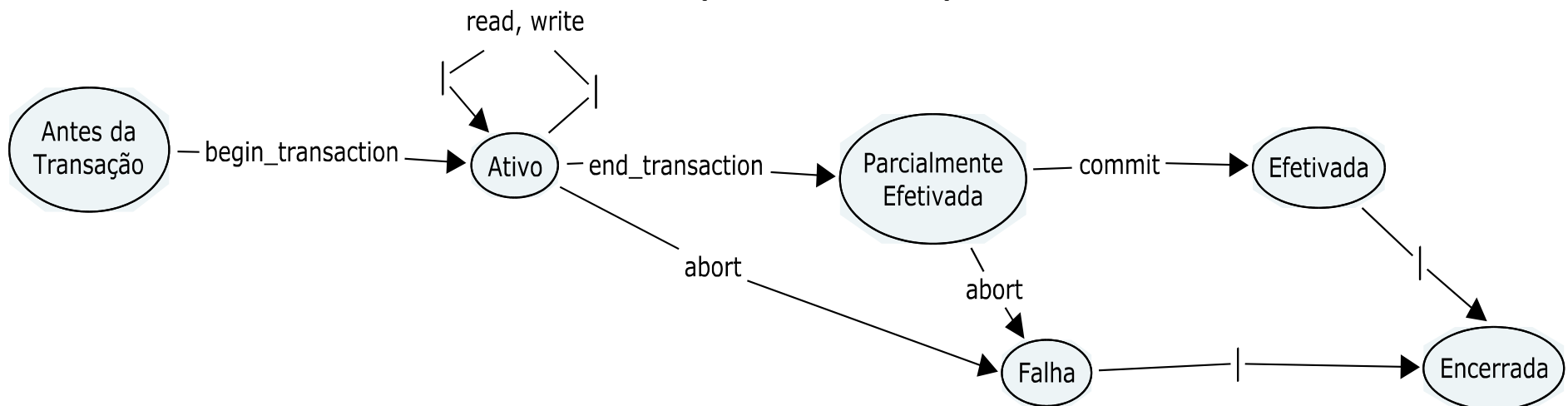
■ READ ou WRITE

■ END_TRANSACTION

■ COMMIT_TRANSACTION

■ ROLLBACK (ou ABORT)

Transições de uma Transação



Exemplo

Transação 1: Transferência

T 1
$ler(X)$
$X = X - N$
$gravar(X)$
$ler(Y)$
$Y = Y + N$
$gravar(Y)$



Prateleira X

transferência



Prateleira Y

Exemplo

Transação 2: Aquisição

T 2

ler (X)

X = X + M

gravar (X)



aquisição

M livros



Prateleira X

Transações Concorrentes

Plano de Execução

■ Necessidade de um Plano de Execução

T1	T2
ler (X) $X = X - N$ gravar (X)	ler (X) $X = X + M$ gravar (X)
ler (Y) $Y = Y + N$ gravar (Y)	

Plano de Execução Serial

T1	T2
ler(X) $X = X - N$ gravar(X) ler(Y) $Y = Y + N$ gravar(Y)	ler(X) $X = X + M$ gravar(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

T1	T2
ler (X) $X = X - N$	
	ler (X) $X = X + M$
gravar (X) ler (Y)	
	gravar (X)
$Y = Y + N$ gravar (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
ler (X) $X = X - N$ gravar (X)	ler (X) $X = X + M$ gravar (X)
ler (Y) $Y = Y + N$ gravar (Y)	

Problema?

T1	T2
ler(X) $X = X - N$	ler(X) $X = X + M$
gravar(X) ler(Y)	gravar(X)
$Y = Y + N$ gravar(Y)	

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
ler (X) $X = X - N$	ler (X) $X = X + M$
gravar (X) ler (Y)	gravar (X)
$Y = Y + N$ gravar (Y)	

■ Sobrescrita de dados alterados sem commit

□ conflito WW (write/write)

(Ramakrishnan, 2003)

Problema?

T1	T2
ler(X) $X = X - N$ gravar(X)	ler(X) $X = X + M$ gravar(X)
ler(Y) ***crash***	

Problema *Dirty Read*

T1	T2
ler (X) $X = X - N$ gravar (X)	ler (X) $X = X + M$ gravar (X)
ler (Y) ***crash***	

- 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
	soma = 0
	ler (A)
	soma = soma + A
	...
ler (X)	
X = X - N	
gravar (X)	
	ler (X)
	soma = soma + X
	ler (Y)
	soma = soma + Y
	...
ler (Y)	
Y = Y + N	
gravar (Y)	

Problema

T1	T3
ler(X) $X = X - N$ gravar(X)	soma = 0 ler(A) soma = soma + A ...
ler(Y) $Y = Y + N$ gravar(Y)	ler(X) soma = soma + X ler(Y) soma = soma + Y ...

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

Problema

T1	T3
ler(X) $X = X - N$ gravar(X)	$soma = 0$ ler(A) $soma = soma + A$... ler(X) $soma = soma + X$ ler(Y) $soma = soma + Y$...
ler(Y) $Y = Y + N$ gravar(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?

T4	T4'
ler (B) verifica(B)	
...	ler (B) verifica(B)
	...
	ler (B) reserva(B)
ler (B) reserva(B)	gravar (B)
gravar (B)	

Problema

Leitura Não Repetitiva

T4	T4'
ler(B) verifica(B)	ler(B) verifica(B)
...	...
ler(B) reserva(B) gravar(B)	ler(B) reserva(B) gravar(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

Consistência DELETE/CASCADE

DELETE FROM Taxi Co
WHERE Taxi.Placa = 'DAE6534';

Cliente (C)

<u>ClId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolksvagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolksvagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999



Corrida (R1)

<u>ClId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

Consistência DELETE/CASCADE

DELETE FROM Taxi Co
WHERE Taxi.Placa = 'DAE6534';

Cliente (C)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolksvagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolksvagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

T1



Corrida (R1)

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

T2

Consistência DELETE/CASCADE

DELETE FROM Taxi Co
WHERE Taxi.Placa = 'DAE6534';

T1 não poderia ser executado sozinho.

Cliente (C)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolksvagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolksvagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

T1



Corrida (R1)

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

T2

Consistência DELETE/CASCADE

DELETE FROM Taxi Co
WHERE Taxi.Placa = 'DAE6534';

Consistência: ambas operações na mesma transação

Cliente (C)

<u>CliId</u>	Nome	CPF
1532	Asdrúbal	448.754.253-65
1755	Doriana	567.387.387-44
1780	Quincas	546.373.762-02



Táxi (TX)



<u>Placa</u>	Marca	Modelo	AnoFab
DAE6534	Ford	Fiesta	1999
DKL4598	Wolksvagen	Gol	2001
DKL7878	Ford	Fiesta	2001
JDM8776	Wolksvagen	Santana	2002
JJM3692	Chevrolet	Corsa	1999

T1



Corrida (R1)

<u>CliId</u>	<u>Placa</u>	<u>DataPedido</u>
1755	DAE6534	15/02/2003
1982	JDM8776	18/02/2003

T1

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 2

T1	T2
ler(X) $X = X - N$ gravar(X) ler(Y) $Y = Y + N$ gravar(Y)	ler(X) $X = X + M$ gravar(X)

Plano Serializável?

T1	T2
ler(X) $X = X - N$	ler(X) $X = X + M$
gravar(X) ler(Y)	gravar(X)
$Y = Y + N$ gravar(Y)	

Não Serializável

T1	T2
ler(X) $X = X - N$	ler(X) $X = X + M$
gravar(X) ler(Y)	gravar(X)
$Y = Y + N$ gravar(Y)	

Plano Serializável?

T1	T2
ler(X) $X = X - N$ gravar(X)	ler(X) $X = X + M$ gravar(X)
ler(Y) $Y = Y + N$ gravar(Y)	

Serializável

T1	T2
ler(X) $X = X - N$ gravar(X)	ler(X) $X = X + M$ gravar(X)
ler(Y) $Y = Y + N$ gravar(Y)	

Plano Serializável
Grafo de Precedência

Grafo de Precedência

Algoritmo

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

(Elmasri, 2010)

Grafo de Precedência

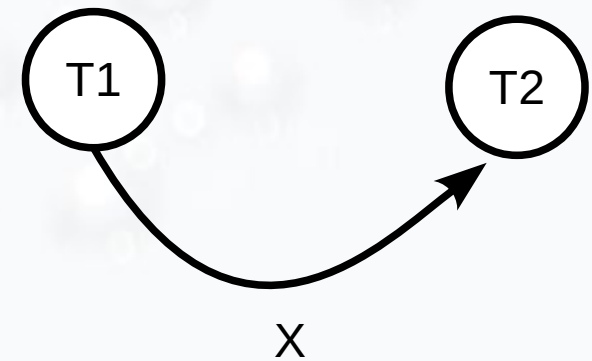
Algoritmo

- Para cada transação crie um nó no grafo
- Para cada caso em S
 - T_i precede e conflita com T_j
 - $aresta(T_i \rightarrow T_j)$
 - Conflita: T_i e T_j realizam uma leitura/gravação no mesmo item e pelo menos um deles é uma gravação
- Serializável \rightarrow sem ciclos

(Ramakrishnan, 2003)

Plano Serial 1

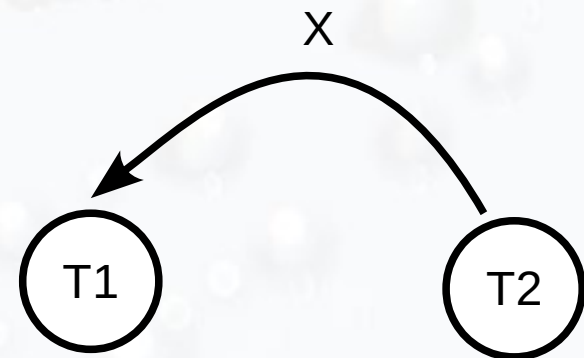
T1	T2
ler (X) $X = X - N$ gravar (X) ler (Y) $Y = Y + N$ gravar (Y)	ler (X) $X = X + M$ gravar (X)



Plano Serial 1

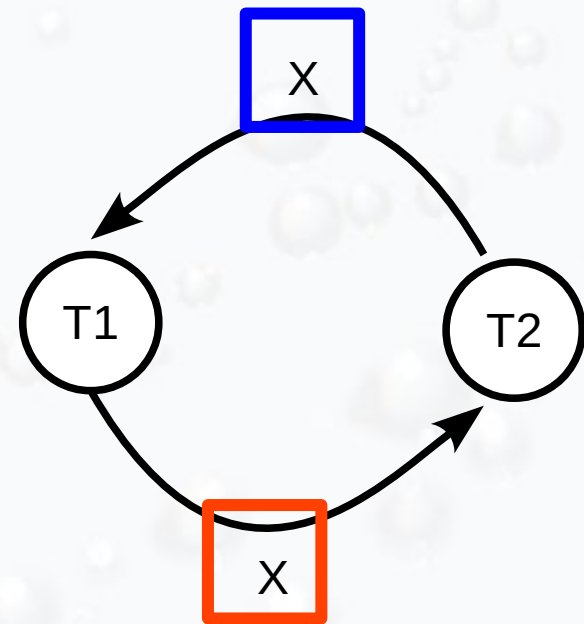
Plano Serial 2

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



Não Serializável

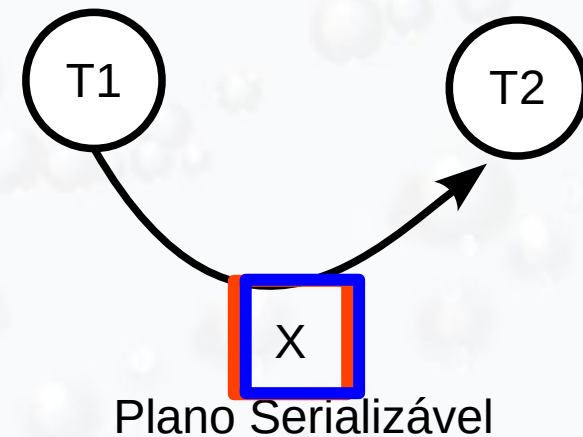
T1	T2
ler(X) $X = X - N$	
	ler(X) $X = X + M$
gravar(X) ler(Y)	
	gravar(X)
$Y = Y + N$ gravar(Y)	



Plano Não Serializável

Serializável

T1	T2
ler(X) $X = X - N$	
gravar(X)	
	ler(X) $X = X + M$
	gravar(X)
ler(Y) $Y = Y + N$	
gravar(Y)	



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

Equivalência por Conflito

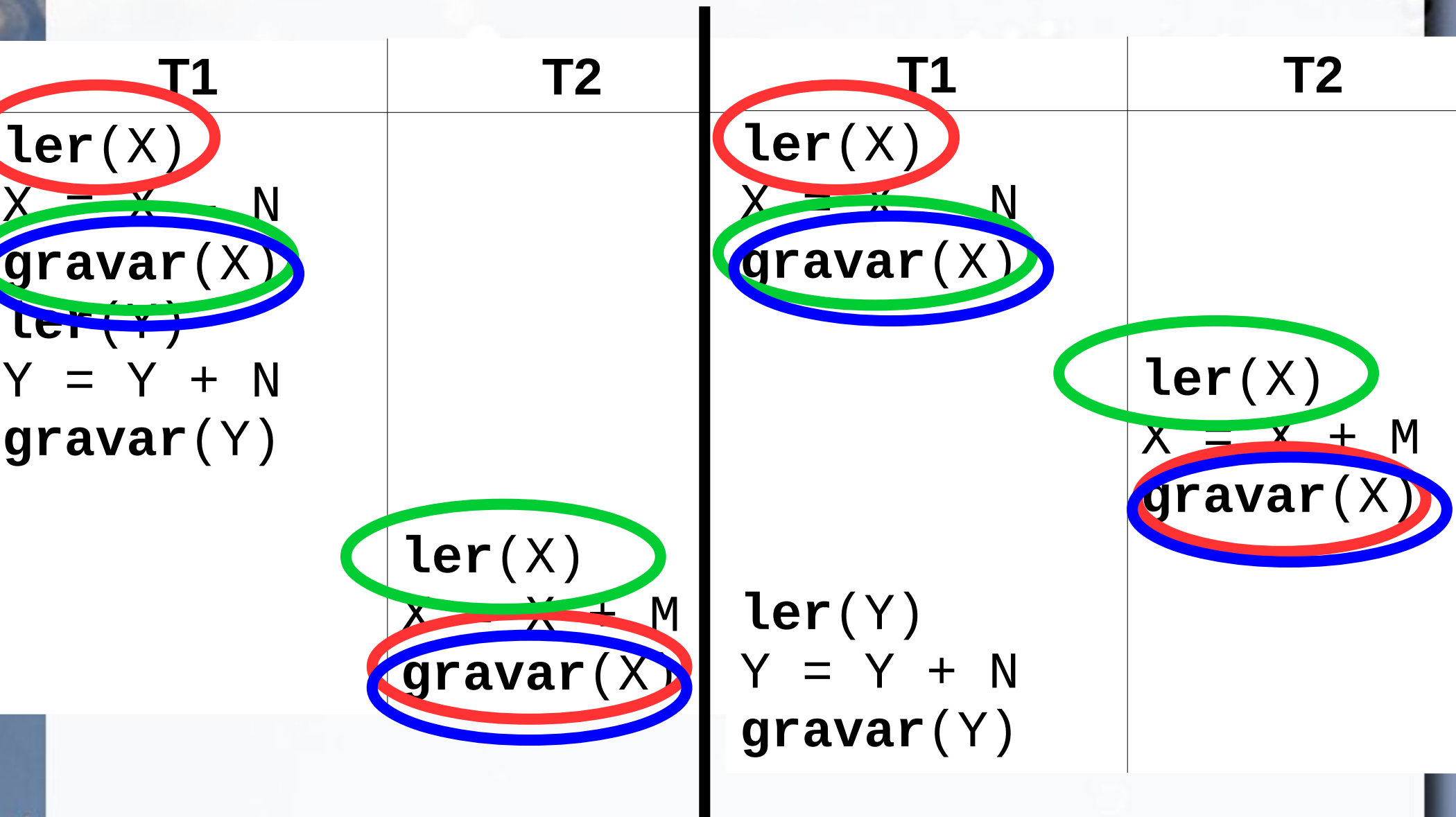
T1	T2	T1	T2
ler(X) $X = X - N$ gravar(X) ler(Y) $Y = Y + N$ gravar(Y)	ler(X) $X = X + M$ gravar(X)	ler(X) $X = X - N$ gravar(X)	ler(X) $X = X + M$ gravar(X)
	ler(Y) $Y = Y + N$ gravar(Y)		

Equivalência por Conflito

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

Equivalência por Conflito

Conflito Serializável



Equivalência por Conflito

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

Não tem Equivalência por Conflito

Não Conflito Serializável

T1	T2	T1	T2
ler(X) $X = X - N$ gravar(X) ler(Y) $Y = Y + N$ gravar(Y)		ler(X) $X = X - N$	
		gravar(X) ler(Y)	ler(X) $X = X + M$
	ler(X) $X = X + M$ gravar(X)		gravar(X)
		$Y = Y + N$ gravar(Y)	

Não tem Equivalência por Conflito

Não Conflito Serializável

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

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

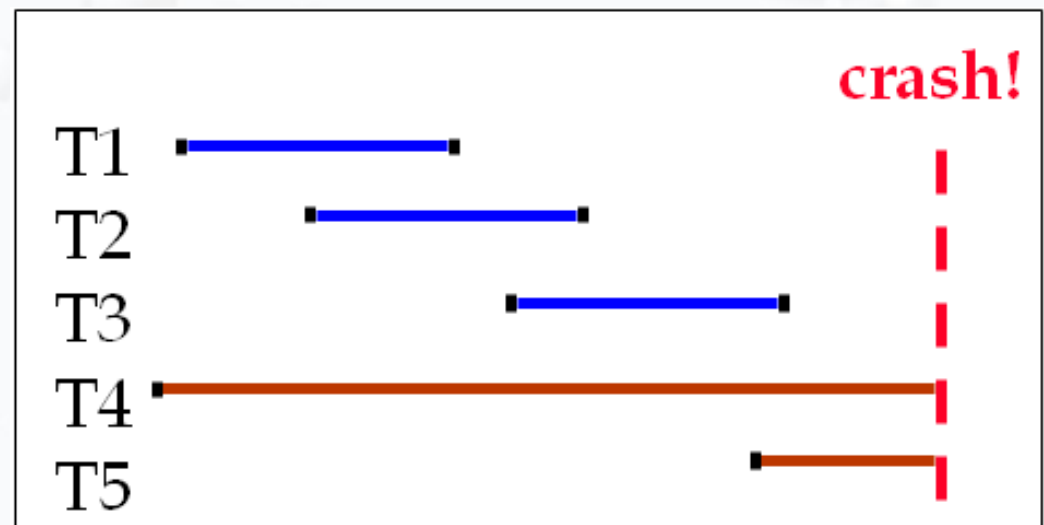
Transação e Atomicidade

■ 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



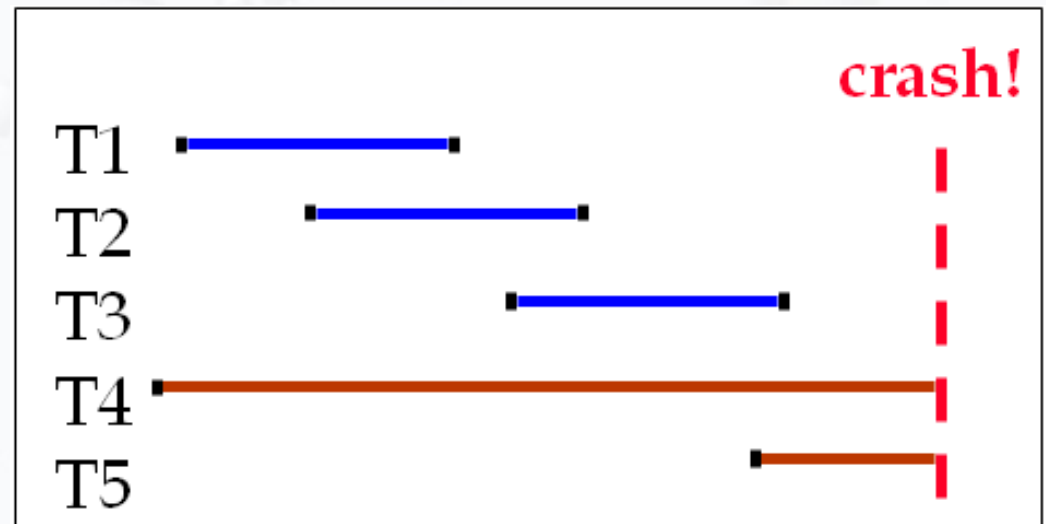
(Ramakrishnan,
2003b)

Transação e Durabilidade

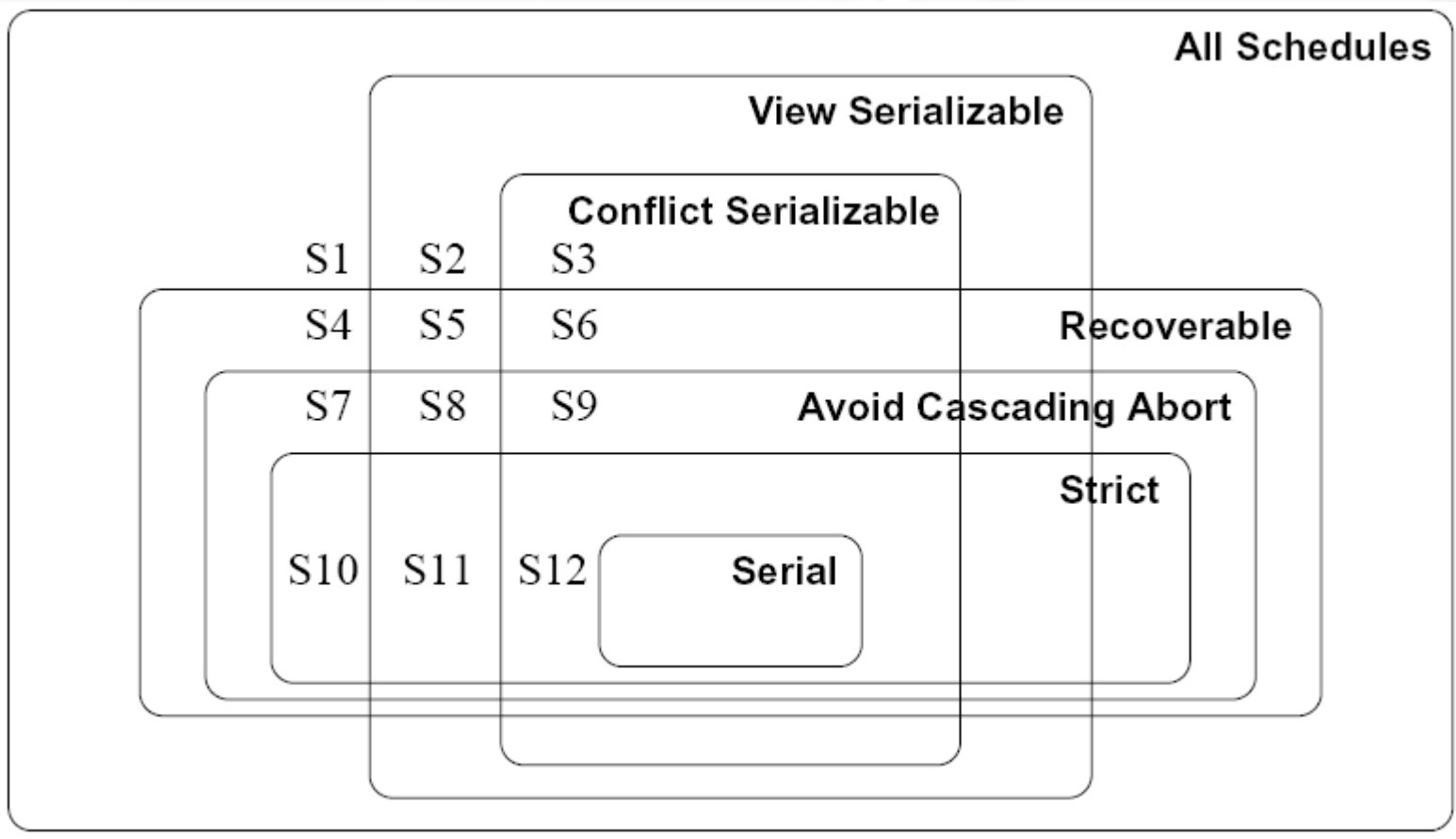
■ O que fazer se o SGBD parar?

■ 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.

Agradecimentos

- Luiz Celso Gomes Jr (professor desta disciplina em 2014) pela contribuição na disciplina e nos slides.
Página do Celso:
<http://dainf.ct.utfpr.edu.br/~gomesjr/>
- Patrícia Cavoto (professora desta disciplina em 2015) pela contribuição na disciplina e nos slides.

André Santanchè

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

Referências

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