

Banco de Dados II

Transações

Controle de Concorrência

Teste para Serialização no Conflito

Dada uma escala S , construa um Grafo de Precedência $G=(V,E)$, onde os vértices V representam as transações pertencentes a S e E é o conjunto de todas as arestas $T_i \rightarrow T_j$ para as quais uma das 3 condições se verifica sobre o mesmo item de dado Q :

1. T_i executa $\text{write}(Q)$ antes de T_j executa $\text{read}(Q)$
2. T_i executa $\text{read}(Q)$ antes de T_j executar $\text{write}(Q)$
3. T_i executa $\text{write}(Q)$ antes de T_j executar $\text{write}(Q)$

Uma aresta $T_i \rightarrow T_j$ no Grafo de Precedência diz que, em qualquer escala serial S' , equivalente a S , T_i deve preceder T_j .

Controle de Concorrência

Equivalência na Visão

Duas escalas $S1$ e $S2$ são equivalentes na visão se, e somente se:

- O conjunto de transações de $S1$ é o mesmo de $S2$;
- Para cada item de dados Q , se a transação T_i lê o valor inicial de Q em $S1$, então T_i também lê o valor inicial de Q em $S2$;
- Para cada item de dados Q , se em $S1$ a transação T_i lê o valor de Q produzido por T_j , então em $S2$ T_i também lê o valor de Q produzido pela transação T_j ;
- Para cada item de dados Q , a transação que **executa** o último write(Q) em $S1$ deve também executar o último write(Q) em $S2$.

Controle de Concorrência

Equivalência na Visão

Exemplo

	T0	T1
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Write(A)
5	Read(B)	
6	Write(B)	
7		Read(B)
8		Write(B)

	T0	T1
1	Read(A)	
2	Write(A)	
3	Read(B)	
4		Read(A)
5		Write(A)
6	Write(B)	
7		Read(B)
8		Write(B)

Escalas **Equivalentes** na Visão

Controle de Concorrência

Equivalência na Visão

Exemplo

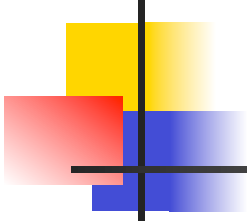
	T0	T1
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Write(A)
5	Read(B)	
6	Write(B)	
7		Read(B)
8		Write(B)

	T0	T1
1	Read(A)	
2	Write(A)	
3	Read(B)	
4		Read(A)
5		Write(A)
6		Read(B)
7	Write(B)	
8		write(B)

Escalas Não Equivalentes na Visão

Controle de Concorrência

Serialização na Visão



Uma escala é **serializável na visão** se
ela é equivalente na visão a
uma escala serial.

Controle de Concorrência

Serialização na Visão

S1

	T0	T1
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Write(A)
5	Read(B)	
6	Write(B)	
7		Read(B)
8		Write(B)

S2

	T0	T1
1	Read(A)	
2	Write(A)	
3	Read(B)	
4	Write(B)	
5		Read(A)
6		Write(A)
7		Read(B)
8		Write(B)

S1 é equivalente na visão à S2 e S2 é serial → S1 é serializável na visão

Controle de Concorrência

Serialização na Visão

Exemplo de Escala Não-Serializável na Visão

	T0	T1
1	Read(Q)	
2		Write(Q)
3	Write(Q)	

Controle de Concorrência

Serialização na Visão

A escala S1 produz o mesmo resultado da escala S2.

S1

	T0	T1
1	Read(A)	
2	$A \leftarrow A - 50$	
5	Write(A)	
6		Read(B)
7		$B \leftarrow B - 10$
8		Write(B)
9	Read(B)	
10	$B \leftarrow B + 50$	
11	Write(B)	
12		Read(A)
13		$A \leftarrow A + 10$
14		Write(A)

S2

	T0	T1
1	Read(A)	
2	$A \leftarrow A - 50$	
3	Write(A)	
4	Read(B)	
5	$B \leftarrow B + 50$	
6	Write(B)	
7		Read(B)
8		$B \leftarrow B - 10$
9		Write(B)
10		Read(A)
11		$A \leftarrow A + 10$
12		Write(A)

Controle de Concorrência

Serialização na Visão

A escala S1 é serializável na visão?

S1

	T0	T1
1	Read(A)	
2	$A \leftarrow A - 50$	
5	Write(A)	
6		Read(B)
7		$B \leftarrow B - 10$
8		Write(B)
9	Read(B)	
10	$B \leftarrow B + 50$	
11	Write(B)	
12		Read(A)
13		$A \leftarrow A + 10$
14		Write(A)

Controle de Concorrência

Serialização na Visão

Exemplo de escala Não-Serializável no Conflito, mas
Serializável na Visão

	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3	Write(Q)		
4			write(Q)

Controle de Concorrência

Teste para Serialização na Visão

- Dada uma escala $S = \{T_0, T_1, T_2, \dots, T_n\}$, considere as transações auxiliares, T_b e T_f , tais que T_b e T_f contêm instruções $\text{write}(Q)$ e $\text{read}(Q)$, respectivamente, para todo item de dado Q acessado em S .
- Construa uma escala S' , inserindo T_b no início de S e inserindo T_f no final de S .

Controle de Concorrência

Teste para Serialização na Visão

- Construa um Grafo de Precedência Rotulado da seguinte forma:

- 1 – Adicione uma aresta $T_i \xrightarrow{0} T_j$ se T_j lê um dado produzido por T_i
- 2 – Remova todas as arestas conectadas a transações inúteis (que não têm caminho para T_f no grafo de precedência)

Controle de Concorrência

Teste para Serialização na Visão

3 – Para cada item de dados Q e transações T_i , T_j e T_k , tais que T_j lê o valor de Q produzido por T_i e T_k executa $\text{write}(Q)$ e $T_k \neq T_i$

faça:

- i) Se $T_i = T_k$ e $T_j \neq T_i$, então insira a aresta $T_j \xrightarrow{0} T_k$
- ii) Se $T_i \neq T_k$ e $T_j = T_i$, então insira a aresta $T_k \xrightarrow{0} T_i$
- iii) Se $T_i \neq T_k$ e $T_j \neq T_i$, então insira as arestas

$T_k \xrightarrow{p} T_i$ e $T_j \xrightarrow{p} T_k$,

onde p é um n° inteiro ainda não utilizado para rotular arestas do grafo

Controle de Concorrência

Teste de Serialização na Visão

Exemplo de Escala Não-Serializável na Visão

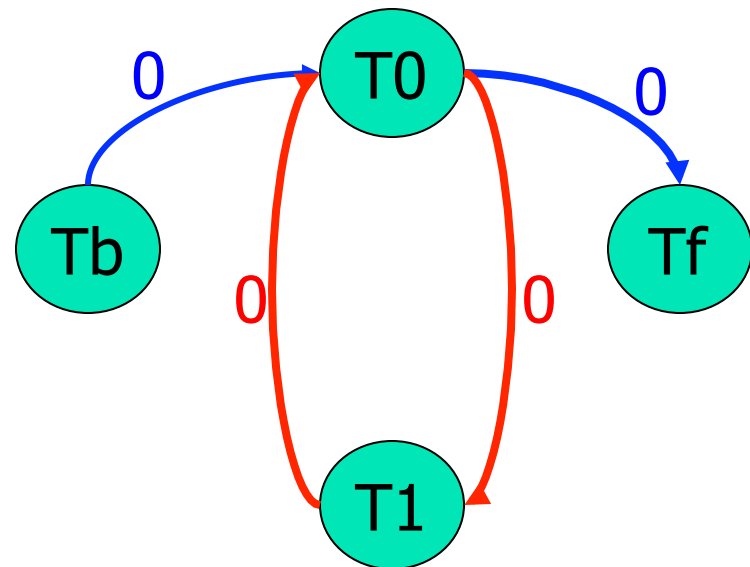
	T0	T1
1	Read(Q)	
2		Write(Q)
3	Write(Q)	

Controle de Concorrência

Teste de Serialização na Visão

Exemplo de Escala Não-Serializável na Visão

	T0	T1
1	Read(Q)	
2		Write(Q)
3	Write(Q)	



Grafo de precedência rotulado é **cíclico**.
Portanto, a escala não é serializável na visão.

Controle de Concorrência

Teste de Serialização na Visão

Exemplo de Escala Serializável na Visão

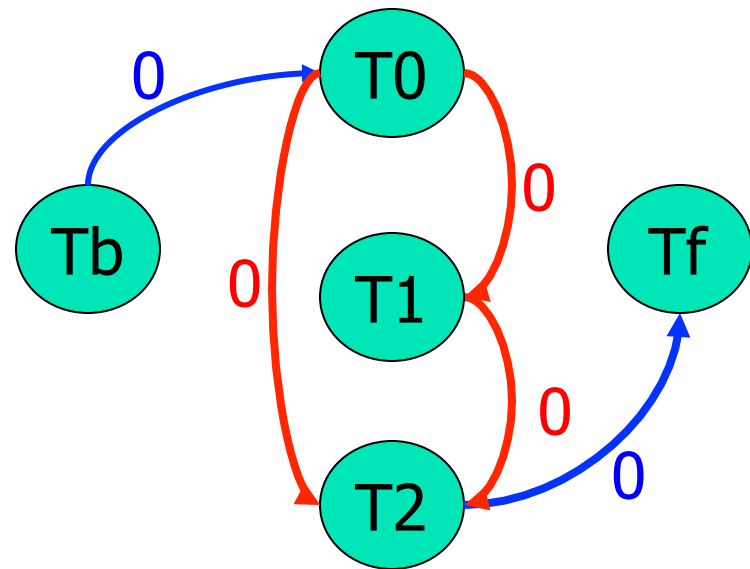
	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3	Write(Q)		
4			write(Q)

Controle de Concorrência

Teste de Serialização na Visão

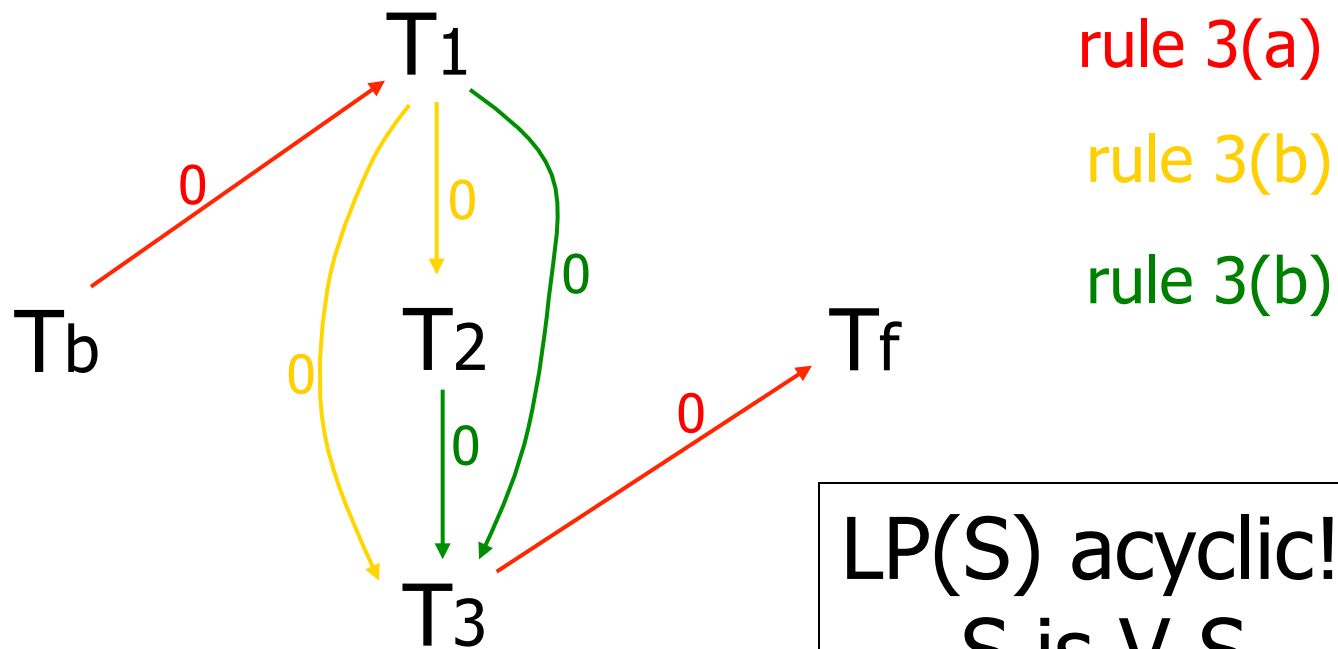
Exemplo de Escala Serializável na Visão

	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3	Write(Q)		
4			write(Q)



Grafo de precedência rotulado é **acíclico**.
Portanto a escala é serializável na visão.

- Example: check if Q is V-S:
- $Q = r1(A) \ w2(A) \ w1(A) \ w3(A)$
- $Q' = wb(A) \Rightarrow r1(A) \ w2(A) \ w1(A) \ w3(A) \Rightarrow rf(A)$



LP(S) acyclic!!
S is V-S

Controle de Concorrência

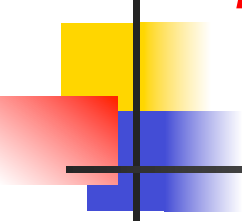
Teste de Serialização na Visão

Esta escala é Serializável na Visão ??

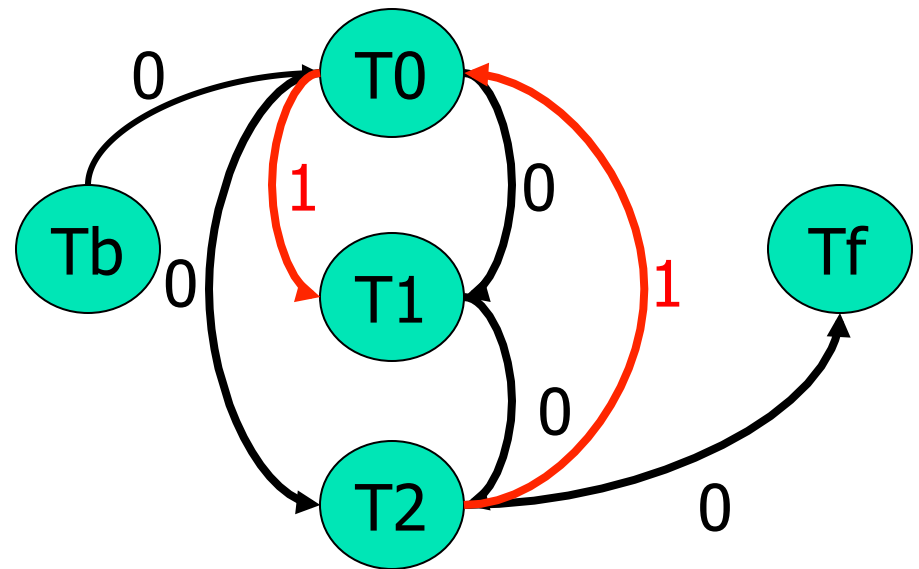
	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3			Read(Q)
4	Write(Q)		
5			Write(Q)

Controle de Concorrência

Teste de Serialização na Visão



	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3			Read(Q)
4	Write(Q)		
5			Write(Q)

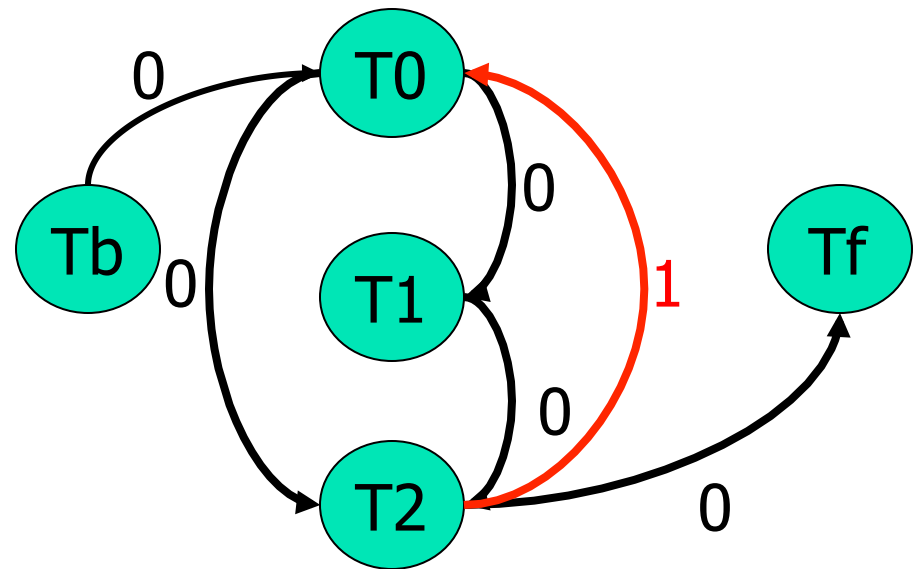


Controle de Concorrência

Teste de Serialização na Visão

Opção 1 → **cíclico**

	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3			Read(Q)
4	Write(Q)		
5			Write(Q)



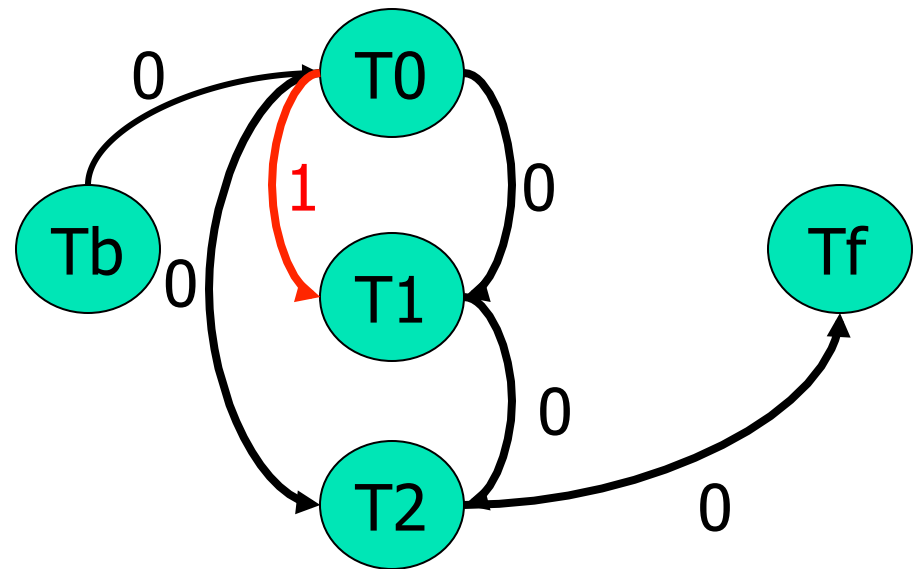
Controle de Concorrência

Teste de Serialização na Visão

Opção 2 → **acíclico**.

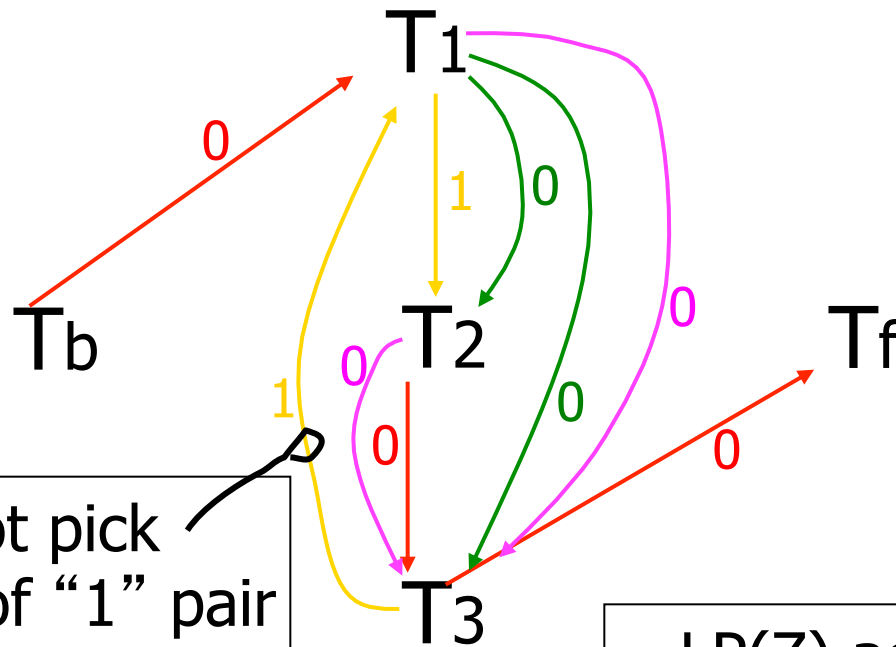
Portanto, **escala** é serializável na visão.

	T0	T1	T2
1	Read(Q)		
2		Write(Q)	
3			Read(Q)
4	Write(Q)		
5			Write(Q)



Another example:

$Z = \text{wb}(A) \Rightarrow \text{r1}(A) \text{ w2}(A) \Rightarrow \text{r3}(A) \text{ w1}(A) \text{ w3}(A) \Rightarrow \text{rf}(A)$



do not pick
this one of "1" pair

LP(Z) acyclic, so Z is V-S
(equivalent to $T_b T_1 T_2 T_3 T_f$)

Controle de Concorrência

Teste de Serialização na Visão

Dados n rótulos diferentes (não considerando o rótulo 0),
existem 2^n possibilidades de grafos.

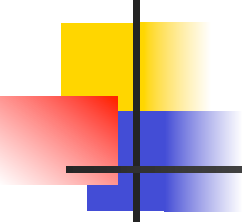
Se pelo menos 1 deles for acíclico, então a escala é
serializável na visão.



Exercicio:

A escala abaixo é serializável? Explique.

$r1(a), r2(b), r1(c), r3(c), r1(b), r3(b), w1(a),$
 $r2(a), r2(c), w2(c), w3(b)$



Schedule 8 (from text) given below produces same outcome as the serial schedule $\langle T_1, T_5 \rangle$, yet is not conflict equivalent or view equivalent to it.

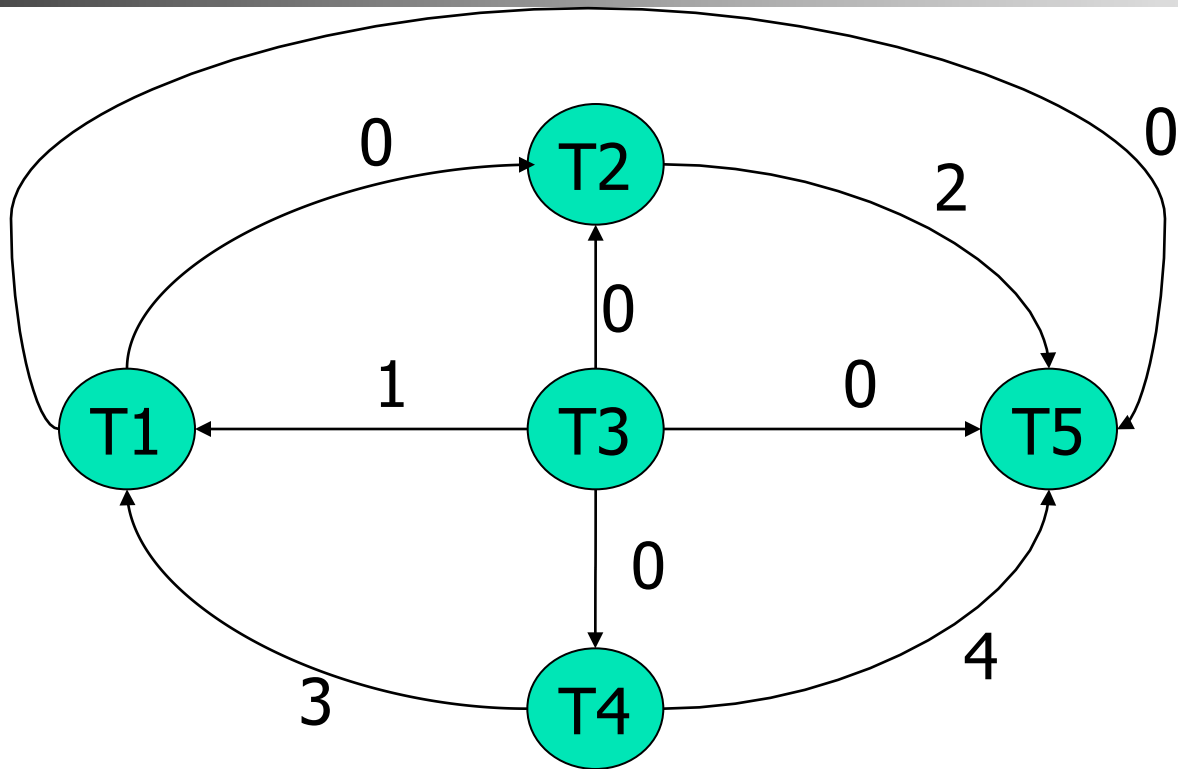
T_1	T_5
read(A) $A := A - 50$ write(A)	
	read(B) $B := B - 10$ write(B)
read(B) $B := B + 50$ write(B)	
	read(A) $A := A + 10$ write(A)

A escala S abaixo é serializável? Explique.

S

T1	T2	T3	T4	T5
		W(Y)		
	R(Y)			
W(X)				
	R(X)			
		W(X)		
			R(X)	
				W(X)

S é serializável na visão!!



T3 → T4 → T1 → T2 → T5

S é equivalente na visão à “T3 → T4 → T1 → T2 → T5”

T1	T2	T3	T4	T5
		W(Y)		
	R(Y)			
W(X) Commit				
	R(X)			
		W(X) CommitC		
	Commit		R(X) Commit	
				W(X) Commit

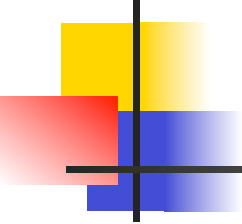
S

T1	T2	T3	T4	T5
		W(Y)		
		W(X)		
			R(X)	
W(X)				
	R(Y)			
	R(X)			
				W(X)

T3 → T4 → T1 → T2 → T5

Controle de Concorrência

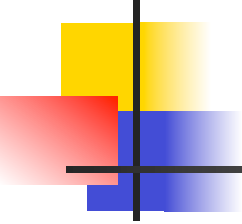
Escalas Recuperáveis



A serialização de escalas garante a consistência do BD quando não há falhas. Entretanto, deve-se garantir esta propriedade também na presença de falhas. Portanto, o SGBD deveria aceitar apenas **escalas recuperáveis**.

Controle de Concorrência

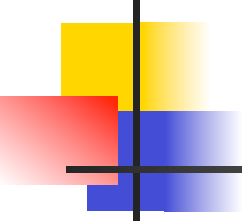
Escalas Recuperáveis



Uma **escala é recuperável** se, para cada par de transações T_i e T_j em que T_j lê um item de dado produzido por T_i , então T_i é efetivada antes de T_j .

Controle de Concorrência

Escalas Recuperáveis



Exemplo de Escala Serializável.

	T0	T1
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Commit
5	Read(B)	
6	Commit	

Controle de Concorrência

Escalas Recuperáveis

Exemplo de Escala Serializável, mas não Recuperável.

T0	T1
1 Read(A)	
2 Write(A)	
3	Read(A)
4	Commit
5 Read(B)	
6 Commit	

T0	T1
1 Read(A)	
2 Write(A)	
3	Read(A)
4	Commit
5 FALHA!!!	
6 Read(B)	

Controle de Concorrência

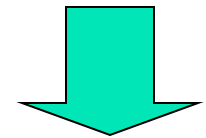
Escalas Recuperáveis

Exemplo de Escala Serializável, mas não Recuperável.

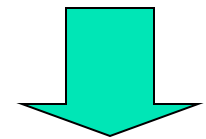
	T0	T1
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Commit
5	Read(B)	
6	Commit	

	T0	T1
1	Read(A)	
2	Write(A)	
3		Read(A)
4		Commit
5	FALHA!	

Rollback (T0)



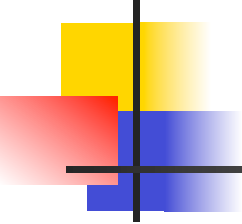
**T1 não pode
ser desfeita.**



**BD inconsistente,
pois T1 leu um
valor impróprio !**

Controle de Concorrência

Escalas Sem Cascata



Uma falha ocorrida em uma transação de uma escala recuperável pode levar ao rollback de muitas transações, o que nem sempre é desejável.

Para se evitar rollbacks em cascata, apenas **escalas sem cascata** deveriam ser aceitas.

Controle de Concorrência

Escalas Sem Cascata

Uma escala de execução sem cascata é aquela na qual para cada par de transações T_i e T_j , tal que T_j lê um item de dados escrito por T_i , a operação de efetivação de T_i aparece antes da operação de leitura de T_j .

Se S é sem cascata, então S é recuperável.



Escalas Estritas

Strict Schedule

A schedule is strict if for any two transactions T1, T2, if a write operation of T1 precedes a *conflicting* operation of T2 (either read or write), then the commit event of T1 also precedes that conflicting operation of T2.

Any strict schedule is cascadeless, but not the converse.



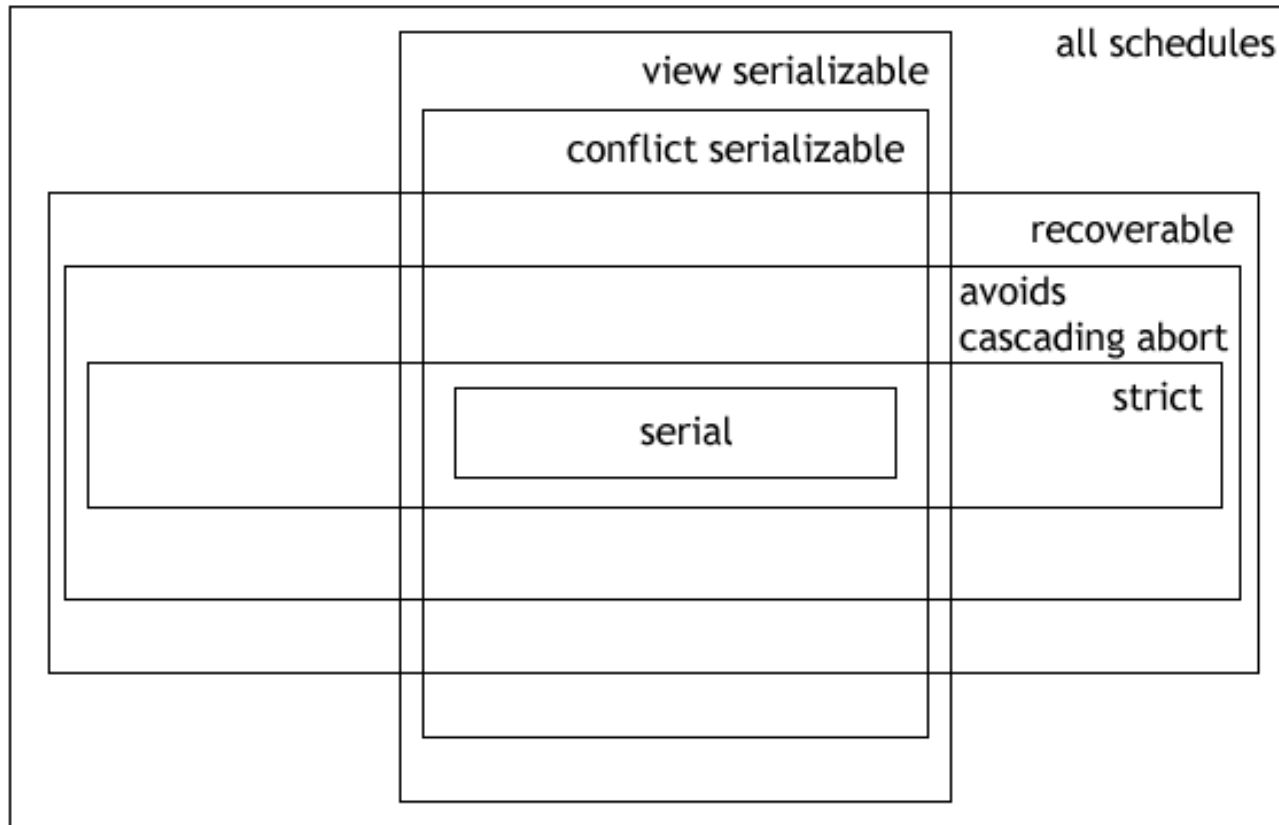
Hierarchical relationship between serializability classes

The following subclass clauses illustrate the hierarchical relationships between serializability classes:

Serial \subset conflict-serializable \subset view-serializable \subset all schedules

Serial \subset strict \subset avoids cascading aborts \subset recoverable \subset all schedules

Hierarchical relationship between serializability classes





Níveis de Isolamento do SQL-92

Serializável

Default do padrão SQL-92.

Read Repetitivo

Leituras efetivadas de registros;
Não permite que T_j atualize um registro entre duas leituras deste registro efetuadas por uma T_i ;
A escala pode não ser serializável.

Read com Efetivação

Apenas registros que sofreram efetivação podem ser lidos;
Não garante leituras repetitivas.

Read sem Efetivação

Permite a leitura de registros que não sofreram efetivação.