

Arthur Barbosa Dornelas Silva - RA: 24.00813-3

Bruno Binaghi Carpi - RA: 24.00246-0

Eduardo Aguiar Leite da Silva - RA: 24.00380-8

1) Acesso concorrente, prós e contras

Vantagens

- Melhor **desempenho/throughput**: mais transações concluídas por unidade de tempo.
- **Menor tempo de espera** percebido por usuários.
- **Melhor utilização** de CPU, disco e caches.

Desvantagens

- **Conflitos** (leitura/escrita e escrita/escrita) exigem controle de concorrência.
- **Interferências** que podem causar leituras sujas, não repetíveis, fantasmas etc.
- **Maior complexidade** do SGBD (bloqueios, versões) e risco de **deadlocks**.

2) Problemas se faltar cada propriedade ACID

- **Atomicidade**: transação parcial deixa o BD em estado “meio aplicado” (ex.: débito feito sem o crédito).

- **Consistência:** regras de integridade violadas (ex.: saldo negativo proibido).
- **Isolamento:** efeitos de **leituras/escritas sujas**, **leituras não repetíveis** e **fantasmas**.
- **Durabilidade:** após “commit”, uma falha faz **perder** atualizações confirmadas.

3-) Conflito-serializabilidade das escalas

Eu gerei as tabelas com os grafos de precedência e o veredito de serializabilidade. Abra as tabelas “Análise de Conflito-Serializabilidade (Q3 e Q4)” na área de resultados acima.

Resumo dos resultados:

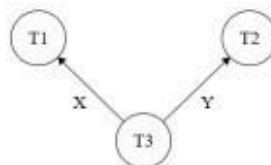
- (i) Serializável por conflito. Arestas: **T3 → T1**, **T3 → T2**. Ordem serial equivalente possível: **T3 → T1 → T2** (ou T3 → T2 → T1).

TABELA POR TEMPO

T1	T2	T3
		W3X
		W3Y
R1X		
	R2Y	
R1Z		
		R3Z
W1X		
	W2Y	

TABELA SERIAL T3 -> T1 -> T2 outra ordem pode ser T3 -> T2 -> T1

T1	T2	T3
		W3X
		W3Y
		R3Z
R1X		
R1Z		
W1X		
	R2Y	
	W2Y	



- (ii) **Não** serializável por conflito. Há ciclo (p.ex., **T2** ↔ **T3** em x e z, e também arestas adicionais em y).

- Conflitos principais:

Em x

$r1(x) \rightarrow w2(x)$

$r1(x) \rightarrow w3(x)$

$w2(x) \rightarrow w3(x)$

$w3(x) \rightarrow r2(x)$

$w2(x) \rightarrow r3(x)$

Em y

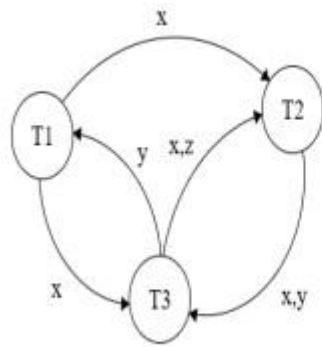
$r2(y) \rightarrow w3(y)$

$w3(y) \rightarrow r1(y)$

Em z

$r3(z) \rightarrow w2(z)$

T1	T2	T3
R1X		
	R2Y	
		R3Z
	W2X	
		W3X
		R3Z
		W3Y
	R2X	
R1Y		
		R3X
	W2Z	



tem ciclo t2 e t3

- (iii) **Não** serializável por conflito. Há ciclos (p.ex., **T2** ↔ **T3** e **T1** ↔ **T3**).

- Conflitos principais:

Em **x**

$r_3(x) \rightarrow w_2(x)$

$w_2(x) \rightarrow w_3(x)$

$w_2(x) \rightarrow r_1(x)$

$w_3(x) \rightarrow r_2(x)$

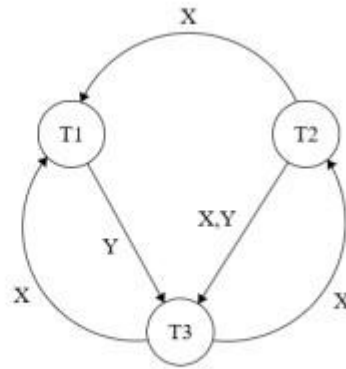
$w_3(x) \rightarrow r_1(x)$

Em **y**

$r_2(y) \rightarrow w_3(y)$

$r_1(y) \rightarrow w_3(y)$

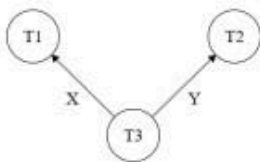
T1	T2	T3
		R3X
	R2Y	
	W2X	
	W2Z	
R1Y		
		W3X
		W3Y
	R2X	
R1X		



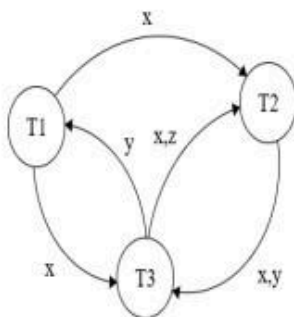
TEM CICLO EM T2 E T3

4) Grafo de precedência

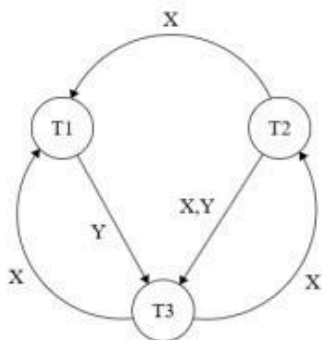
1) Conflito-serializável. Ordens seriais possíveis: $T3 \rightarrow T1 \rightarrow T2$ ou $T3 \rightarrow T2 \rightarrow T1$.



2) Grafo: existe ciclo $T2 \rightleftharpoons T3 \rightarrow$ não conflito-serializável



3) Grafo: ciclo $T2 \rightleftharpoons T3 \rightarrow$ não conflito-serializável.



5) Escala(s) equivalente(s) para as serializáveis

Na tabela “Q5 – Escalas equivalentes (serial) para as serializáveis” eu montei uma escala serial equivalente para (i).

Exemplo (uma opção válida respeitando a ordem **T3** → **T1** → **T2** e a ordem interna de cada transação):

w3(x), w3(y), r3(z), r1(x), r1(z), w1(x), r2(y), w2(y)

- TABELA SERIAL T3 -> T1 -> T2

T1	T2	T3
		W3X
		W3Y
		R3Z
R1X		
R1Z		
W1X		
	R2Y	
	W2Y	

- TABELA SERIAL T3 -> T2 -> T1

T1	T2	T3
		W3X
		W3Y
		R3Z
	R2Y	
	W2Y	
R1X		
R1Z		
W1X		

6) Para as não serializáveis: uma escala sob 2PL

Para (ii) e (iii), como há ciclo, não existe escala **equivalente por conflito**. Sob **2PL estrito** (crescimento de locks e depois liberação), uma execução **serial** é garantida. Na tabela “Q6 – Escalas geradas por 2PL para as não-serializáveis” eu propus, por simplicidade, a ordem **T1 → T2 → T3** e reescrevi cada escala agrupando por transação (mantendo a ordem interna de cada T). Essas são execuções válidas que 2PL pode produzir.

2) **T1 → T2 → T3**

T1	T2	T3
LOCK X-X	LOCK X-X	LOCK X-X
LOCK X-Y	LOCK X-Y	LOCK X-Y
R1X	LOCK X-Z	LOCK X-Z
R1Y	R2Y	R3Z
UNLOCK-X	W2X	W3X
UNLOCK-Y	R2X	R3Z
INICIA O T2	W2Z	W3Y
	UNLOCK-X	R3X
	UNLOCK-Y	UNLOCK-X
	UNLOCK-Z	UNLOCK-Y
	INICIA T3	UNLOCK-Z

3) T2 → T3 → T1

T2	T3	T1
LOCK X-X	LOCK X-X	LOCK X-X
LOCK X-Y	LOCK X-Y	LOCK X-Y
LOCK X-Z	R3X	R1Y
R2Y	W3X	R1X
W2X	W3Y	UNLOCK-X
W2Z	UNLOCK-X	UNLOCK-Y
R2X	UNLOCK-Y	
UNLOCK-X	INICIA O T1	
UNLOCK-Y		
UNLOCK-Z		
INICIA O T3		