



**UNIVERSIDADE FEDERAL DO
MARANHÃO - UFMA**

Banco de Dados II

Processamento de Transações II

Carlos Eduardo Portela Serra de Castro

*

Processamento de Transações II

Sumário

Definindo Plano de Execução

Baseado na Restaurabilidade

Baseado na Seriabilidade

Suporte a Transações em SQL

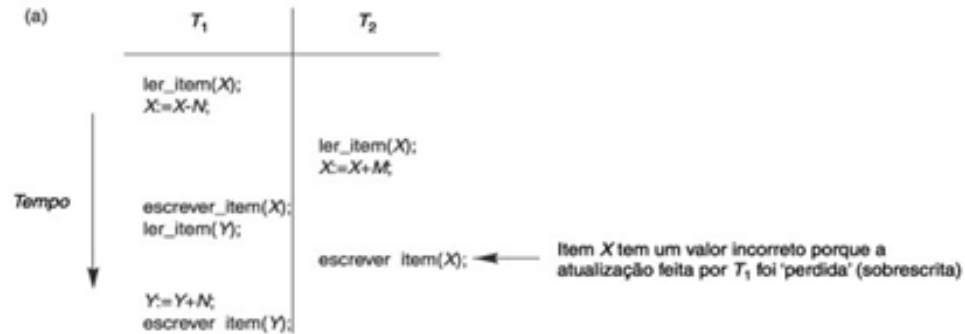
Definindo Plano de Execução Baseado na Restaurabilidade

(Schedules Recoverability)

Planos de Execução (Histórico) de Transações.

- A ordem de execução das operações de várias transações executando concorrentemente na forma Intercalada.
- Um plano S de n transações T_1, T_2, \dots, T_n é a ordenação das operações dessas transações, sujeito à uma restrição, tal que, para cada transação T_i em S , as operações de T_i em S devem aparecer na mesma ordem que elas ocorrem em T_i .

Plano de Execução para o problema da atualização perdida



Sa : r1(X); r2(X); w1(X); r1(Y); w2(X); w2(y);

r: read, w: write, c: commit; a;abort

Operações Conflitantes em um escalonamento

Duas operações são ditas em conflito se:

- 1 - Pertencem a diferentes transações;
- 2 - Acessam o mesmo item X;
- 3 - Pelo menos uma das operações é um escrever_item(X).

Plano completo

Um plano é chamado de plano completo se:

- 1 – As operações em S são exatamente as operações de T_1, T_2, \dots, T_n , tendo um *commit* ou um *abort* como última operação de cada transação no plano;
- 2 – Para quaisquer pares de operação da mesma transação T_i , sua ordem de aparecimento em S será a mesma que em T_i ;
- 3 – Para quaisquer operações conflitantes, uma das duas precisa aparecer antes de outra no plano.

Plano restaurável (recuperável)

Um plano S é restaurável se nenhuma transação T em S for efetivada até que todas as transações T' que tiverem gravado um item lido por T , tenham sido efetivadas.

- Plano restaurável:

Sa': r1(X); r2(X); w1(X); r1(Y); w2(X); c2; w1(Y); c1;

- Plano não restaurável:

Sc: r1(X); w1(X); r2(X); r1(Y); w2(X); c2; a1;

T2 lê X de T1 e é efetivada antes de T1.

- Reversão em Cascata (*cascading rollback*)

Uma transação não efetivada tem que ser revertida porque lê um item de uma transação que falhou.

- Plano livre de reversão em Cascata (*avoid cascading rollback*)

Cada transação do plano lê somente itens que foram gravados por transações efetivadas.

- Plano Restrito (*restrict schedule*)

Nenhuma transação pode ler ou escrever itens até que a última transação que grave X tenha ainda sido efetivada ou abortada

Definindo Plano de Execução Baseado na Serialidade

(Schedules Serializability)

- Serializabilidade de Escalonamentos

Sem intercalação, se duas transações são submetidas, então todas as operações da transação T1 (em seqüência) são seguidas por todas as operações de T2, ou vice-versa.

- Teoria da Serializabilidade

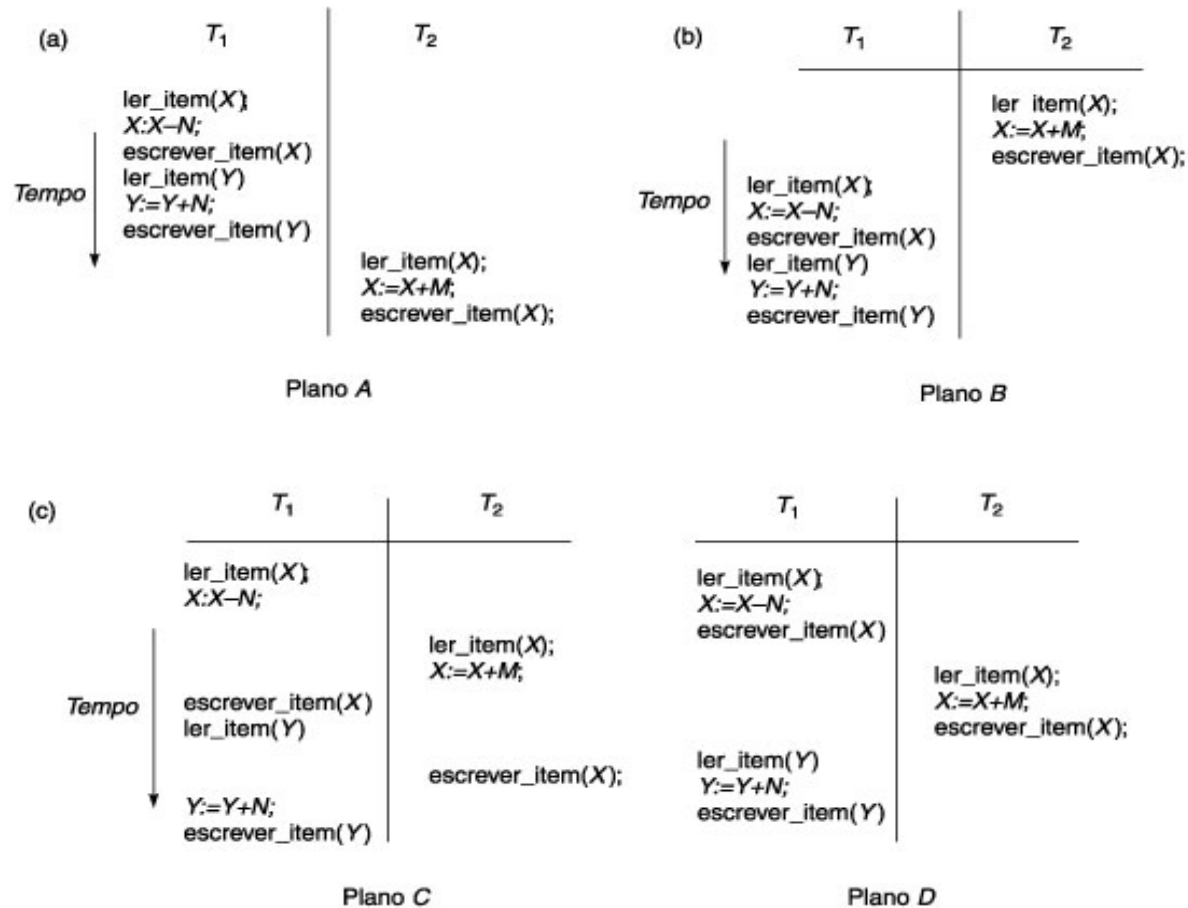
Tenta determinar quais escalonamentos são “corretos” e quais não são e desenvolver técnicas que permitam apenas escalonamentos corretos.

Exemplos de planos seriais e não-seriais envolvendo as transações T_1 e T_2 .

(a) Plano serial A: T_2 , seguido por T_1 .

(b) Plano serial B: T_2 seguido por T_1 .

(c) Dois planos não-seriais, C e D, com intercalação de operações.



Planos Seriais, Não-Seriais e de Conflitos Serializáveis

Escalonamento Serial

- As operações de cada transação são executadas consecutivamente, sem operações intercaladas com as de outras transações

Escalonamento Não-serial

- As operações de uma transação são executadas intercaladas com operações de outra transação

Escalonamento Serializável

- É equivalente a algum escalonamento serial com as mesmas 'n' transações

Escalonamento de Resultados Equivalentes

- Produzem o mesmo resultado final do BD

Dois planos que têm resultados equivalentes quando o valor inicial de $X = 100$, mas que, no geral, não são resultados equivalentes.

S_1	S_2
<code>ler_item(X);</code>	<code>ler_item(X);</code>
<code>X:=X+10;</code>	<code>X:=X*1.1;</code>
<code>escrever_item(X);</code>	<code>escrever_item(X);</code>

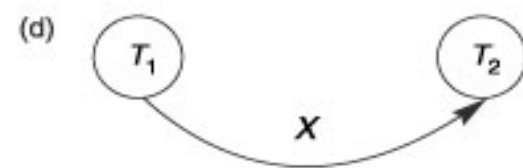
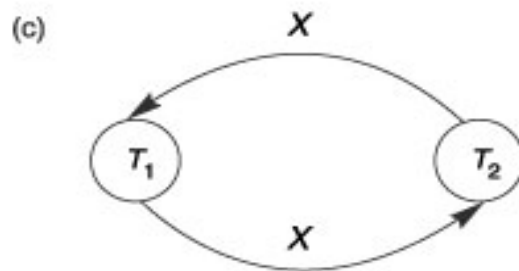
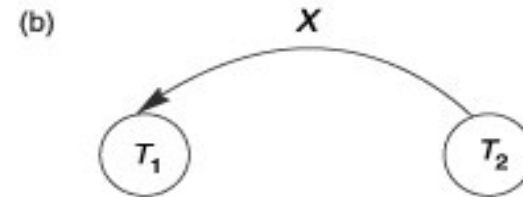
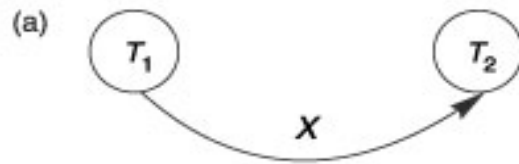
Testando o Conflito de Serialidade de um Plano

Grafo de Precedência: grafo dirigido $G=(N,E)$ que consiste de um conjunto de nós $N=\{T1, T2, \dots, Tn\}$ e um conjunto de setas dirigidas $E=\{e1, e2, \dots, em\}$ onde cada nó representa uma transação do plano.

*(Grafo de Precedência ou Grafo de Serialização)

Construindo os grafos de precedência para teste de conflito serialidade dos planos A a D da Figura 17.5.

- (a) Grafo de precedência para o plano serial A.
- (b) Grafo de precedência para o plano serial B.
- (c) Grafo de precedência para o plano C (não serializável).
- (d) Grafo de precedência para o plano D (serializável, equivalente ao plano A).



Teste de Conflito de Serialidade

Algoritmo

1. Para cada transação participando de um escalonamento S , criar um nó rotulado T_i no grafo de precedência
2. Para cada caso em S onde uma transação T_j executa um `ler_item(x)` após um `escrever_item(x)` executado por outra transação T_i crie uma seta ($T_i \rightarrow T_j$) no grafo de precedência

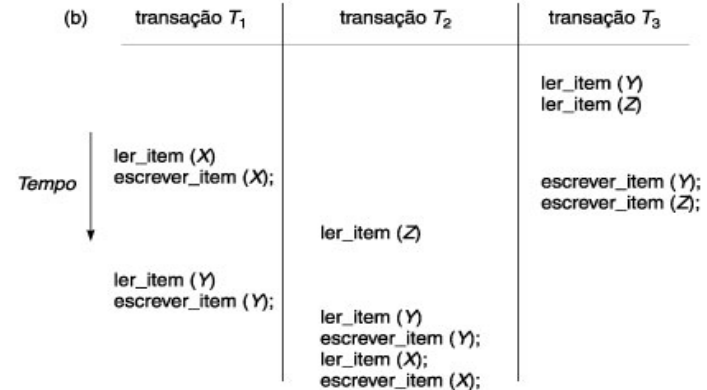
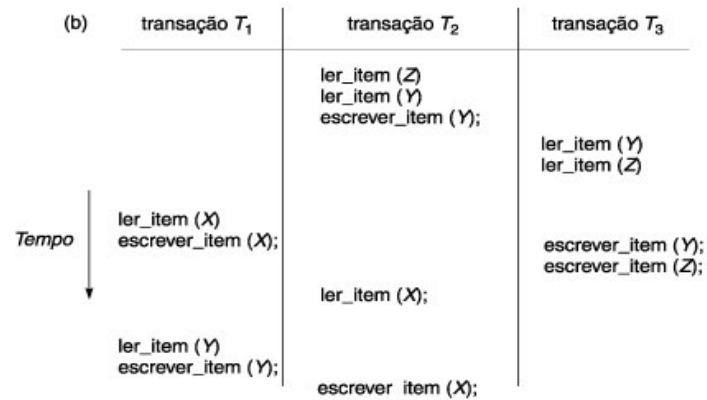
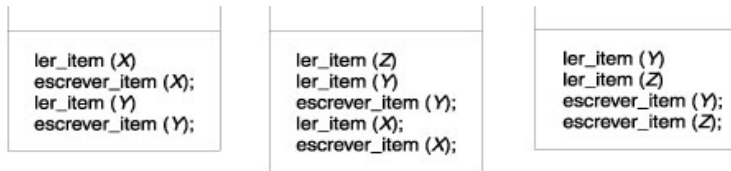
Teste de Conflito de Serialidade

- 3. Para cada caso em S onde uma transação T_j executa um `escrever_item(x)` após um `ler_item(x)` executado por outra transação T_i crie uma seta ($T_i \rightarrow T_j$) no grafo de precedência
- 4. Para cada caso em S onde uma transação T_j executa um `escrever_item(x)` após um `escrever_item(x)` executado por outra transação T_i crie uma seta ($T_i \rightarrow T_j$) no grafo de precedência
- 5. Um plano S será serializável se e somente se o grafo de precedência não tem ciclos

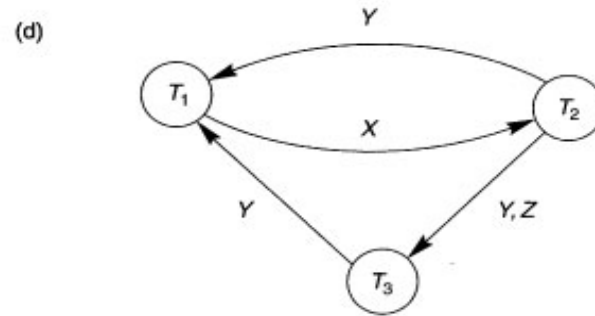
Aplicações de Serialidade

- Plano serial x Plano serializável

Outro exemplo de teste de serialidade. (a) As operações READ e WRITE das três transações T_1 , T_2 , e T_3 . (b) Plano E . (c) Plano F .



Outro exemplo de teste de serialidade. (d) Grafo de precedência para o plano E. (e) Grafo de precedência para o plano F. (f) Grafo de precedência com dois planos seriais equivalentes.

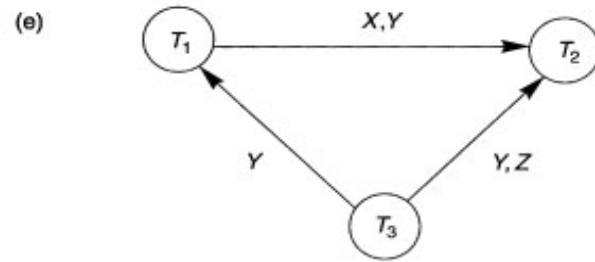


Planos seriais equivalentes

Nenhuma

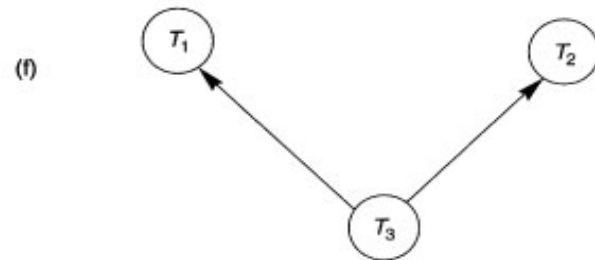
Razão

ciclo $X(T_1 \rightarrow T_2), Y(T_2 \rightarrow T_1)$
 ciclo $X(T_1 \rightarrow T_2), YZ(T_2 \rightarrow T_3), Y(T_3 \rightarrow T_1)$



Planos seriais equivalentes

$T_3 \rightarrow T_1 \rightarrow T_2$



Planos seriais equivalentes

$T_3 \rightarrow T_1 \rightarrow T_2$

$T_3 \rightarrow T_2 \rightarrow T_1$

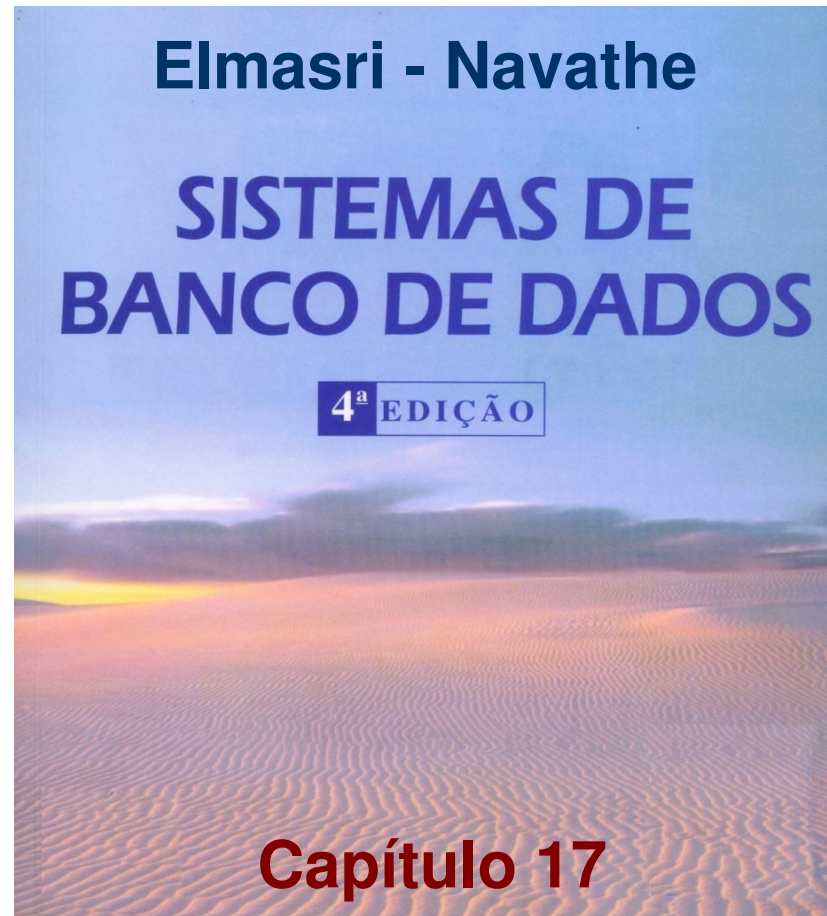
Suporte de Transações SQL

- Conceito similar ao já definido
- Não existe `Begin_Transaction` explícita
- Modo de acesso:
 - `Read_only`
 - `Reas_Write`
- Nível de Isolamento:
 - `Read uncommitted`
 - `Read committed`
 - `Repeatle read`
 - `Serializable`

Tabela 21.1

Violações possíveis com base nos níveis de isolamento definidos na SQL.

Tipo de violação			
Nível de isolamento	Leitura suja	Leitura não repetitiva	Fantasma
READ UNCOMMITTED	Sim	Sim	Sim
READ COMMITTED	Não	Sim	Sim
REPEATABLE READ	Não	Não	Sim
SERIALIZABLE	Não	Não	Não



Atenção:

Leitura do

Capítulo 17

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