

Lista de Exercícios I: Processamento de Transações

A atividade consiste na realização de exercícios presentes no capítulo "Introdução aos Conceitos e Teoria do Processamento de Transações" do livro texto da disciplina (R. Elmasri, R.; Navathe, S.B. *Sistemas de Banco de Dados*).

Perguntas de revisão

- 21.1. O que significa a execução concorrente de transações de banco de dados em um sistema multiusuário? Discuta por que o controle de concorrência é necessário e dê exemplos informais.
- 21.2. Discuta os diferentes tipos de falhas. O que significa uma falha catastrófica?
- 21.3. Discuta as ações tomadas pelas operações `read_item` e `write_item` em um banco de dados.
- 21.5. Para que é usado o log do sistema? Quais são os tipos característicos de registros em um log do sistema? O que são pontos de confirmação da transação e por que eles são importantes?
- 21.6. Discuta as propriedades de atomicidade, durabilidade, isolamento e preservação da consistência de uma transação de banco de dados.
- 21.7. O que é um schedule (histórico)? Defina os conceitos de schedules recuperáveis, sem cascata e estritos, e compare-os em matéria de sua facilidade de recuperação.
- 21.9. O que é um schedule serial? O que é um schedule serializável? Por que um schedule serial é considerado correto? Por que um schedule serializável é considerado correto?
- 21.11. Discuta como a serialização é usada para impor o controle de concorrência em um sistema de banco de dados. Por que a serialização às vezes é considerada muito restritiva como uma medida da exatidão para os schedules?
- 21.12. Descreva os quatro níveis de isolamento em SQL.
- 21.13. Defina as violações causadas por cada um dos seguintes itens: leitura suja, leitura não repetitiva e fantasmas.

Exercícios

- 21.14. Mude a transação T_2 da Figura 21.2(b) para
`read_item(X);`
`X := X + M;`
`if X > 90 then exit`
`else write_item(X);`
 Discuta o resultado final dos diferentes schedules na Figura 21.3(a) e (b), onde $M = 2$ e $N = 2$, em relação às seguintes questões: a inclusão da condição acima muda o resultado final? O resultado obedece à regra de consistência implícita (de que a capacidade de X é 90)?
- 21.15. Repita o Exercício 21.14, acrescentando uma verificação em T_1 de modo que Y não exceda 90.
- 21.16. Inclua o `commit` da operação ao final de cada uma das transações T_1 e T_2 na Figura 21.2, e depois liste todos os schedules possíveis para as transações modificadas. Determine quais dos schedules são recuperáveis, quais são sem cascata e quais são estritos.
- 21.20. Por que uma instrução de fim de transação explícita é necessária em SQL, mas não uma instrução de início explícita?
- 21.21. Descreva situações em que cada um dos diferentes níveis de isolamento seriam úteis para o processamento de transação.
- 21.22. Qual dos seguintes schedules é serializável (de conflito)? Para cada schedule serializável, determine os schedules seriais equivalentes.
 - a. $r_1(X); r_3(X); w_1(X); r_2(X); w_3(X);$
 - b. $r_1(X); r_3(X); w_3(X); w_1(X); r_2(X);$
 - c. $r_3(X); r_2(X); w_3(X); r_1(X); w_1(X);$
 - d. $r_3(X); r_2(X); r_1(X); w_3(X); w_1(X);$
- 21.23. Considere as três transações T_1 , T_2 e T_3 , e os schedules S_1 e S_2 a seguir. Desenhe os grafos de serialização (precedência) para S_1 e S_2 e indique se cada schedule é serializável ou não. Se um schedule for serializável, escreva o(s) schedule(s) serial(is) equivalente(s).

$T_1: r_1(X); r_1(Z); w_1(X);$
 $T_2: r_2(Z); r_2(Y); w_2(Z); w_2(Y);$
 $T_3: r_3(X); r_3(Y); w_3(Y);$
 $S_1: r_1(X); r_2(Z); r_1(Z); r_3(X); r_3(Y); w_1(X);$
 $w_3(Y); r_2(Y); w_2(Z); w_2(Y);$
 $S_2: r_1(X); r_2(Z); r_3(X); r_1(Z); r_2(Y); r_3(Y); w_1(X);$
 $w_2(Z); w_3(Y); w_2(Y);$

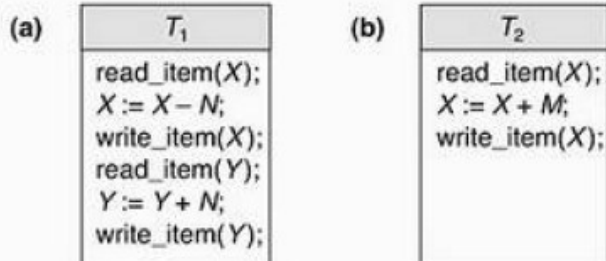


Figura 21.2

Duas transações de exemplo. (a) Transação T_1 . (b) Transação T_2 .

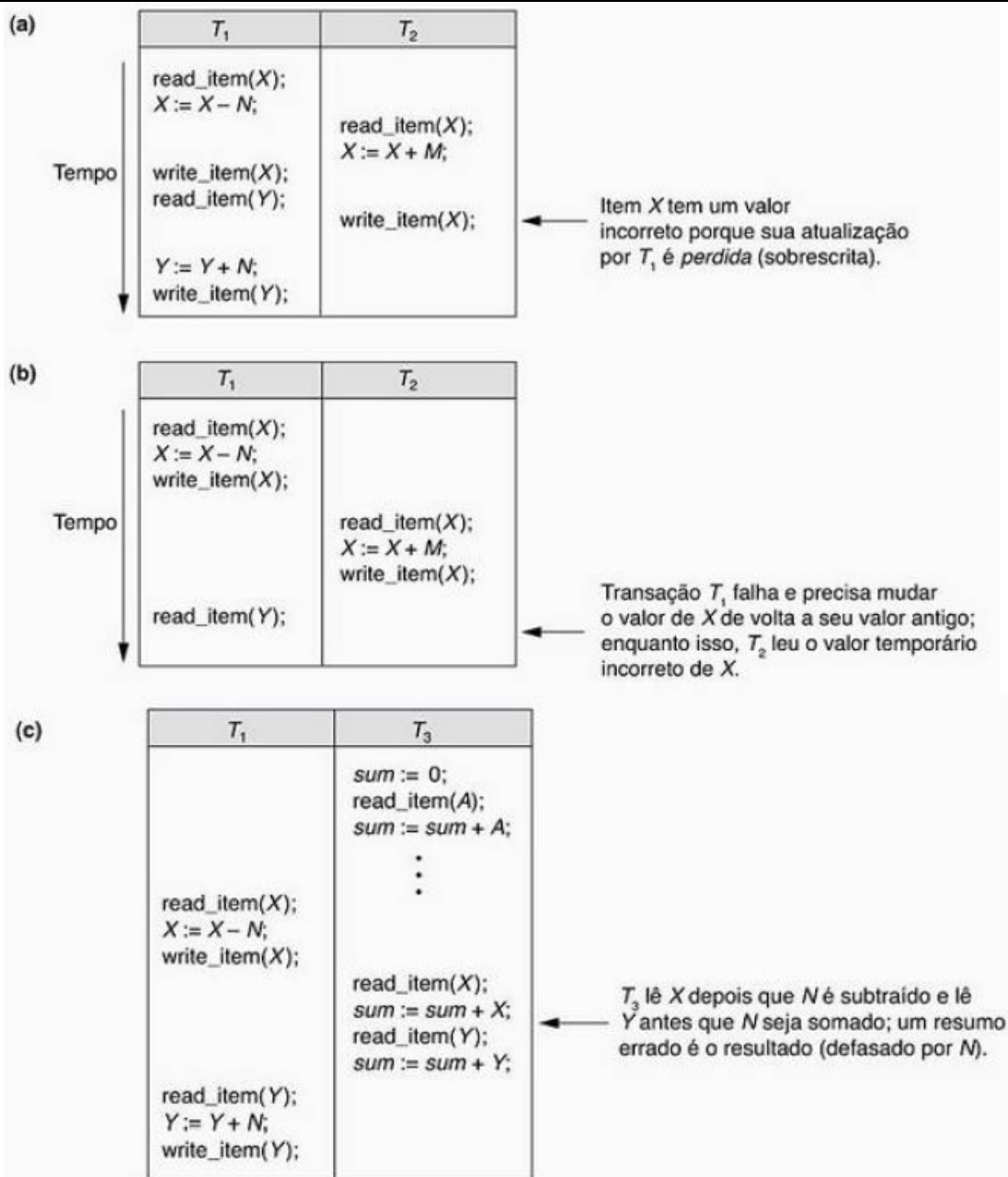


Figura 21.3

Alguns problemas que ocorrem quando a execução simultânea não é controlada. (a) O problema da atualização perdida. (b) O problema da atualização temporária. (c) O problema do resumo incorreto.