



Sistemas de Gerência de Banco de Dados

Exercício sobre Logs

Questão para exercitar

- Considerando a utilização do algoritmo de UNDO/REDO para a recuperação de falhas e como técnica de atualização imediata do BD
- E considerando um arquivo de *log*, em que temos as seguintes premissas:
 - um comando de instrução(T_A , start) indica o início de uma transação T_A
 - uma instrução (T_A , commit) indica o registro de conclusão da transação
 - X, Y, Z, W, K e L indicam os itens de dados alterados pelas transações

Questão para exercitar (continua)

- Então, no comando (T_A , X , 20, 50), temos respectivamente que:
 - T_A significa o identificador de transação
 - X significa o item afetado
 - 20 representa o valor antigo
 - 50 representa o novo valor

Questão para exercitar (continua)

Marcação sequencial de data e hora (*timestamping*) de execução das instruções

1. (T_A , *start*)
2. (T_A , X, 20, 50)
3. (T_B , *start*)
4. (T_B , Y, 40, 50)
5. (T_A , Z, 56, 34)
6. (T_B , W, 65, 33)
7. (T_B , *commit*)
8. *checkpoint*
9. (T_C , *start*)
10. (T_A , K, 22, 34)
11. (T_C , L, 44, 55)
12. (T_C , *commit*)
13. *crash*

Marcação de ponto de verificação (*checkpoint*)

Marcação de ponto de falha (*crash*)

Questão para exercitar (continua)

T_A deverá ser desfeita (**UNDO**), em função de não ter concluído ainda (não há instrução **commit**)

T_B não precisa ser feito **nada**, pois instrução de **commit** está antes do checkpoint

1. (T_A , start)
2. (T_A , x, 20, 50)
3. (T_B , start)
4. (T_B , y, 40, 50)
5. (T_A , z, 56, 34)
6. (T_B , w, 65, 33)
7. (T_B , commit)
8. checkpoint
9. (T_C , start)
10. (T_A , k, 22, 34)
11. (T_C , l, 44, 55)
12. (T_C , commit)
13. crash

T_C deverá ser refeita (**REDO**), pois há a instrução de **commit**, mas ainda não havia passado por um checkpoint



Sistemas de Gerência de Banco de Dados

Sistemas de Gerência de Banco de Dados

