

Projeto de Banco de Dados



Sistema de Recuperação

PROF. DR. THIAGO ELIAS

Sistema de Recuperação



- Um sistema de computador está sujeito a falhas.
- Assim, um sistema de BD deve precaver-se para que as propriedades de atomicidade e durabilidade sejam preservadas, mesmo ocorrendo falhas.

Classificação das Falhas



- Falha de transação
 - Erro Lógico: entrada inadequada, dado não encontrado...
 - Erro de Sistema: exemplo seria o deadlock
- Queda do sistema
- Falha no disco

Estrutura de Armazenamento



- Tipos de Armazenamento:
 - Volátil
 - Não-volátil
 - Estável

Implementação do Armazenamento Estável

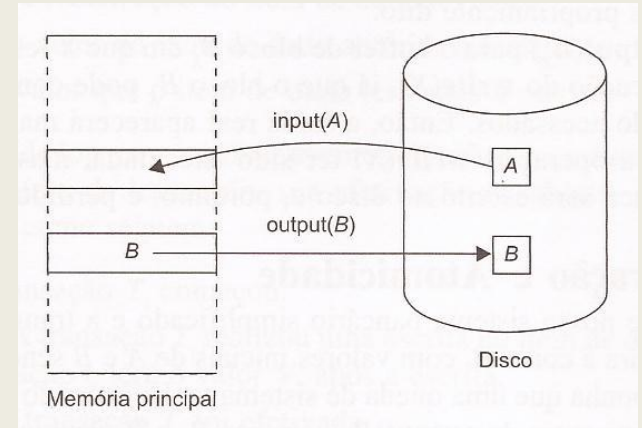


- Para implementar o armazenamento estável é necessário replicar a informação em vários meios de armazenamento não-volátil.
 - Exemplo: Sistemas RAID

Acesso aos Dados



- Input(B): transfere o *bloco físico* B para a memória principal.
- Output(B): transfere o *bloco de buffer* B para o disco.
- Read(X): atribui o valor de X a uma variável local.
 - Se o bloco de X não estiver na MP, é executado um input(X).
 - Designa o valor de X à variável.
- Write(X): o contrário de read(X).
 - Se o bloco de X não estiver na MP, é feito um input(X).
 - Designa o valor da variável à X na MP.



OBS: Ambas as operações não forçam o output. Este pode acontecer depois, ou por falta de espaço na MP ou decisão do Sistema de BD.

Recuperação e Atomicidade



- Considerando o exemplo anterior da transferência de valores entre duas contas e, imaginando uma falha entre o output(A) e output(B), qual a solução para o problema?
 - Reexecutar as transações?
 - Não reexecutar?
- Em ambos os casos, o BD permanece inconsistente.
- Solução?
 - Primeiro mandar para o armazenamento estável/não-volátil as informações que descrevem todas as modificações realizadas pela transação (antes de modificar o BD em si)
- Em seguida, iniciaremos o estudo de técnicas que possibilitam essa tarefa.

Recuperação Baseada em Log



- O Log é uma sequência de registro de log que mantém um arquivo atualizado das atividades no BD.
- Um registro de log possui os seguintes campos:
 - Identificador da transação
 - Identificador do item de dado
 - Valor antigo
 - Valor novo
- Também existem os registros que marcam o início e término (efetivada ou abortada) da transação:
 - Start
 - Commit
 - Abort

Recuperação Baseada em Log

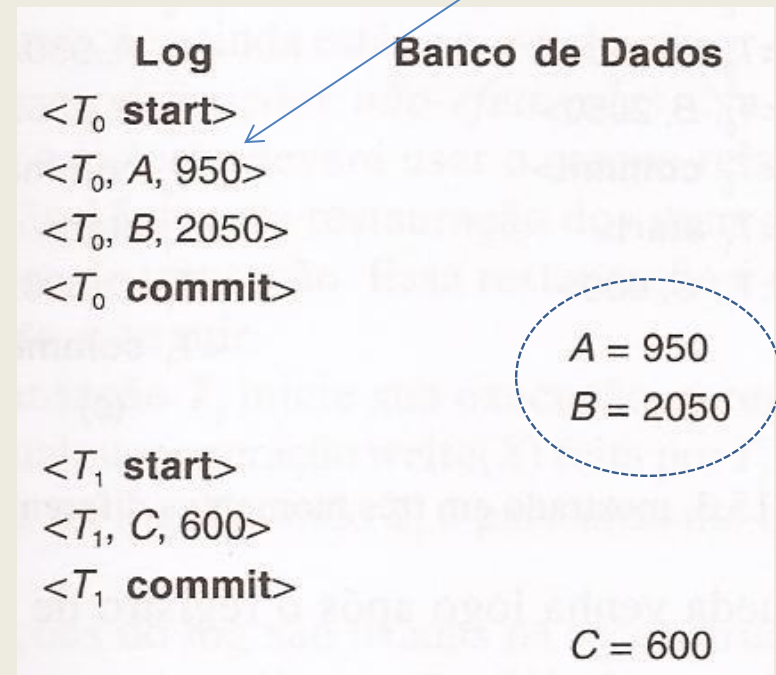


- Para que os registro de Log sejam úteis, estes devem residir numa memória estável.
- Será assumido, por enquanto, que os Log's serão armazenamos imediatamente após a sua criação.
- Problema:
 - O Log pode ocupar um espaço absurdamente grande, uma vez que registra toda a atividade no BD

Recuperação Baseada em Log com Modificações ADIADAS do BD



- O que é “Modificação Adiada”?
- A transação entra no estado efetivado apenas depois que todos os registros de log são armazenados na memória estável.
- Considere duas transações: uma transfere 50 da conta A para a conta B e a outra saca 100 da conta C.
- Os logs podem ser vistos ao lado.
- Por que não há o valor antigo do item de dado?!!!!



OBS: Em caso de falha, apenas um REDO precisará ser feito!!

Recuperação Baseada em Log com Modificações ADIADAS do BD



- O que fazer com as situações a seguir?

$\langle T_0 \text{ start} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$

(a)

$\langle T_0 \text{ start} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$
 $\langle T_0 \text{ commit} \rangle$
 $\langle T_1 \text{ start} \rangle$
 $\langle T_1, C, 600 \rangle$

(b)

$\langle T_0 \text{ start} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$
 $\langle T_0 \text{ commit} \rangle$
 $\langle T_1 \text{ start} \rangle$
 $\langle T_1, C, 600 \rangle$
 $\langle T_1 \text{ commit} \rangle$

(c)

Recuperação Baseada em Log com Modificações IMEDIATAS do BD

- As modificações no BD acontecem enquanto as transações ainda estão no estado ativo.
- Essas escritas são chamadas de *modificações não-efetivadas*.
- Observe os logs das mesmas transações anteriores.
- Por que há o valor antigo do item de dado?!!!!

Log	Banco de Dados
$\langle T_0, \text{start} \rangle$	
$\langle T_0, A, 1000, 950 \rangle$	
$\langle T_0, B, 2000, 2050 \rangle$	
	A = 950
	B = 2050
$\langle T_0, \text{commit} \rangle$	
$\langle T_1, \text{start} \rangle$	
$\langle T_1, C, 700, 600 \rangle$	
	C = 600
$\langle T_1, \text{commit} \rangle$	

OBS: Em caso de falha, o REDO ou o UNDO precisarão ser executados!!

Recuperação Baseada em Log com Modificações IMEDIATAS do BD



- O que fazer com as situações a seguir? (CORRIGIR!!!!)

$\langle T_0 \text{ start} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$

(a)

$\langle T_0 \text{ start} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$
 $\langle T_0 \text{ commit} \rangle$
 $\langle T_1 \text{ start} \rangle$
 $\langle T_1, C, 600 \rangle$

(b)

$\langle T_0 \text{ start} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$
 $\langle T_0 \text{ commit} \rangle$
 $\langle T_1 \text{ start} \rangle$
 $\langle T_1, C, 600 \rangle$
 $\langle T_1 \text{ commit} \rangle$

(c)

Checkpoints

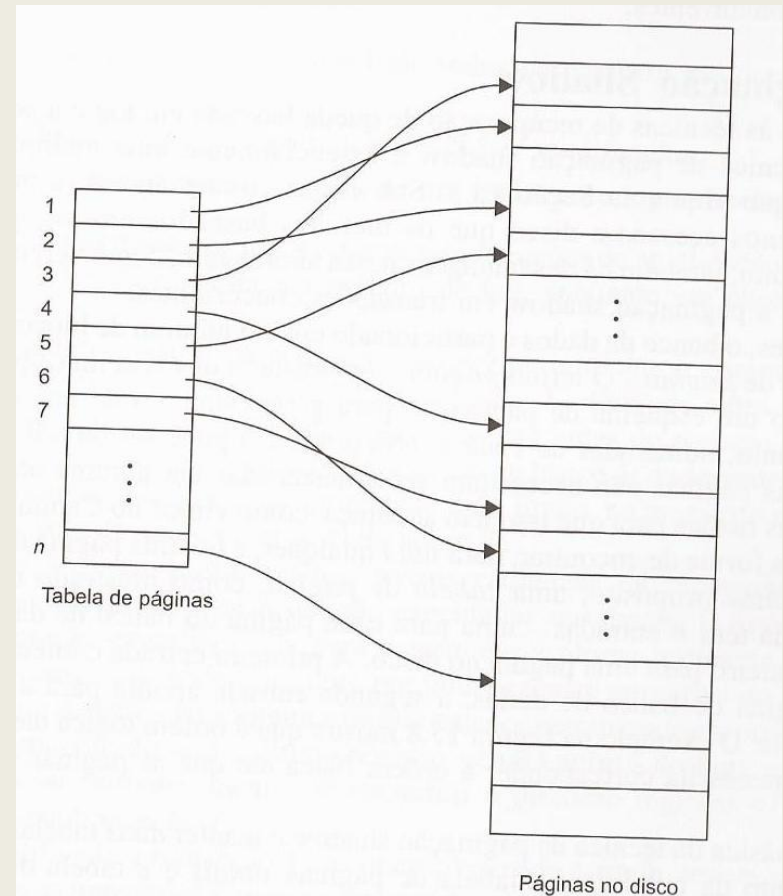


- Em caso de falha, o sistema deve varrer todo o arquivo de Log. Problemas:
 - Tempo.
 - Recuperar transações que já foram efetivadas.
- Os checkpoints são escritos periodicamente e exigem a execução da seguinte sequência:
 - Saída para armazenamento estável de todos os registros residentes na memória principal.
 - Saída da memória principal de todos os blocos de buffer modificados.
 - Saída para armazenamento estável de um registro de log <checkpoint>
- Não é permitida nenhuma execução de atualização do BD durante o registro do checkpoint.

Paginação Shadow



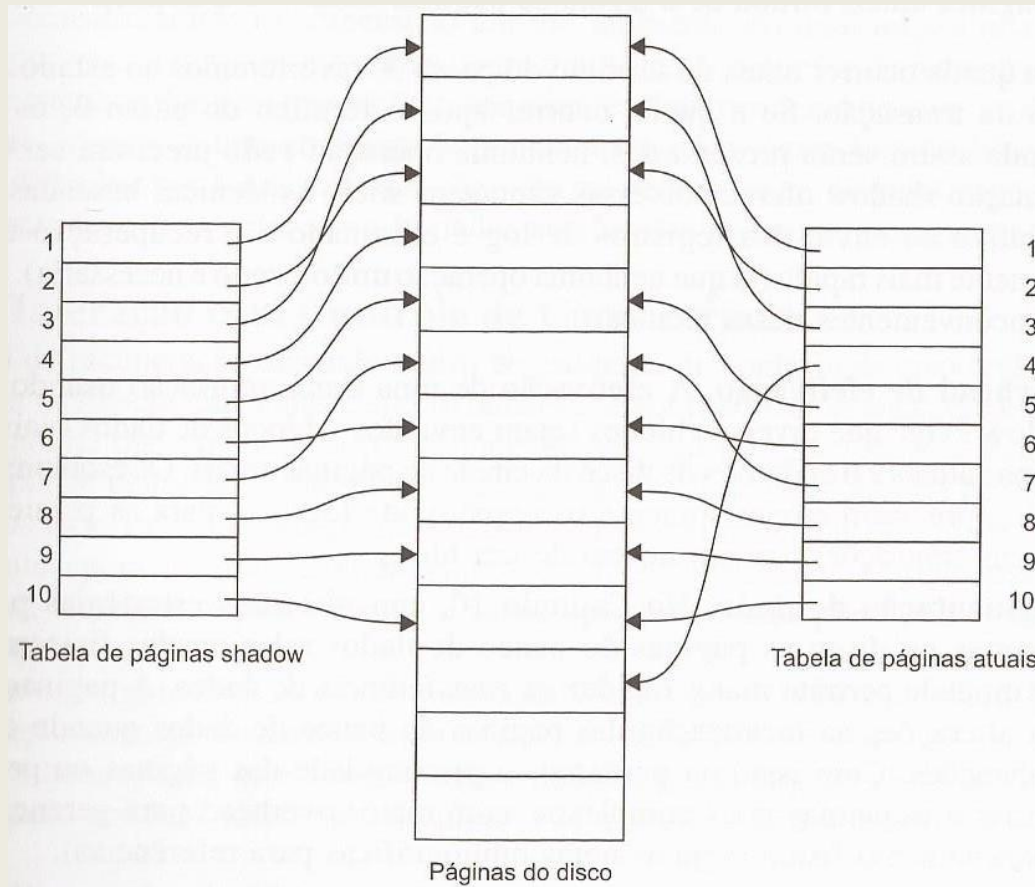
- Evolução da técnica Cópia Shadow
- Vantagem
 - Menos acesso ao disco
- Desvantagem
 - Difícil aplicação em transações concorrentes
- Uso de tabelas de páginas



Paginação Shadow



- Como funciona a alteração de itens de dados no BD?



Paginação Shadow



- Desvantagens:
 - Overhead de efetivação: para a efetivação de uma única transação, há a manipulação de diversos blocos de memória.
 - Fragmentação de dados
 - Coleta de lixo