

## TÉCNICAS DE CONTROLE DE CONCORRÊNCIA - parte 1 -

Profa. Dra. Maria Madalena Dias

1

## TÉCNICAS DE CONTROLE DE CONCORRÊNCIA - parte 1 -

- Introdução
- Protocolos de Controle de Concorrência
  - Protocolos Pessimistas
  - Protocolos Otimistas
- Protocolos de Controle de Concorrência Baseados em Bloqueios
  - Bloqueio Binária
  - Bloqueio Exclusivo e Compartilhado

2

### Introdução

- Técnicas utilizadas para garantir a *isolação* de transações concorrentes
- Garantem a serialização dos escalonamentos através da utilização de protocolos
- Protocolos de controle de concorrência:
  - oferecem várias regras que, se seguidas pelas transações, garantem a serialização de todos os escalonamentos nos quais as transações participam

3

### Protocolos de Controle de Concorrência

#### • Pessimistas



- baseados na premissa que conflitos entre transações ocorrem com frequência
- (alta probabilidade)

#### Otimistas



baseados na premissa que conflitos entre transações são raros (baixa probabilidade)

4

### Protocolos de Controle de Concorrência

#### • Pessimistas



- testam as transações antes da execução de suas operações
- *two phase locking*  
*timestamp ordering*

#### Otimistas



testam as transações após a execução de suas operações

↓

técnicas de validação

5

### Protocolos Pessimistas

- *Two Phase Locking (2PL)*
  - solução padrão para o problema de controle de concorrência em SGBDs convencionais
  - protocolo baseado em bloqueios
- *Timestamp Ordering*
  - protocolo baseado no *timestamp* das transações
  - *timestamp*
    - identificador único associado a cada transação
  - valores de *timestamp* são associados às transações respeitando-se a ordem na qual estas são submetidas ao sistema

6

## Protocolos Otimistas

- Técnicas de Validação
  - também chamadas de técnicas de certificação ou otimistas
  - transação em uma técnica de validação
    - deve possuir duas ou três fases
      - fase de leitura
      - fase de validação
      - fase de escrita (opcional)
    - fase de leitura
      - a transação pode ler valores dos itens de dados do BD; no entanto, atualizações são aplicadas somente a cópias locais dos itens de dados mantidos no *workspace* da transação

7

## Técnicas de Validação

- Fase de validação
  - realiza uma checagem para garantir que a serialização não será violada se as atualizações da transação forem aplicadas ao BD
- Fase de escrita
  - se a fase de validação obtiver sucesso,
    - então as atualizações são aplicadas ao BD
    - senão as atualizações são descartadas e as transações reinicializadas
  - problema
    - caso haja muito conflito entre as transações, então muitas atualizações serão descartadas e muitas transações serão reinicializadas

8

## Protocolos de Controle de Concorrência Baseados em Bloqueios

- Bloqueio (*lock*)
  - variável associada a um dado do BD
  - indica o *status* do item de dado com relação às possíveis operações que podem ser nele aplicadas
  - geralmente existe uma trava associada a cada item de dado do BD
  - especifica a sincronização de acesso aos itens de dado do BD por transações concorrentes

9

## Bloqueio Binário

- Pode assumir dois *status* ou valores
  - *locked* : 1 : bloqueado
  - *unlocked* : 0 : desbloqueado
- Associação ao item de dado
  - < item\_dado, bloqueio > + fila de espera
- se bloqueio ( item\_dado ) = 1
  - então item\_dado não pode ser acessado
- se bloqueio ( item\_dado ) = 0
  - então item\_dado pode ser acessado

10

## Bloqueio Binário

- Operações adicionais
  - *lock\_item* ( item\_dado )
  - *unlock\_item* ( item\_dado )
- Operações *lock\_item* e *unlock\_item*
  - são unidades indivisíveis
  - utilizam conceitos semelhantes ao de região crítica em Sistemas Operacionais
- SGBD
  - oferece um subsistema de gerenciamento de bloqueios para manter e controlar o acesso aos bloqueios

11

## Bloqueio Binário

- **Força exclusão mútua**
  - apenas uma transação por vez pode possuir o bloqueio de um determinado item de dado
  - duas transações não podem acessar o mesmo item de dado concorrentemente
- Conceito muito restritivo

12

## Regras Seguidas pelas Transações

- uma transação T deve realizar a operação *lock\_item(x)* antes de qualquer operação *read\_item(x)* ou *write\_item(x)* de T
- uma transação T deve realizar a operação *unlock\_item(x)* depois de todas as operações *read\_item(x)* ou *write\_item(x)* de T

13

## Bloqueio Exclusivo e Compartilhado

- Bloqueio de modo múltiplo
  - *read\_lock(x), write\_lock(x), unlock(x)*
- Bloqueio compartilhado - *read\_lock(x)*
  - várias transações podem acessar o mesmo item de dado simultaneamente
  - finalidade: leitura
- Bloqueio exclusivo - *write\_lock(x)*
  - apenas uma transação pode acessar um item de dado em um determinado tempo
  - finalidade: escrita

14

## Bloqueio Exclusivo e Compartilhado

- Pode assumir três *status* ou valores
  - *read\_locked* : bloqueado para leitura
  - *write\_locked* : bloqueado para escrita
  - *unlocked* : desbloqueado
- Associação ao item de dado
  - *< item\_dado, bloqueio, nro\_leitores >*
    - + fila de espera para transações que não podem acessar o item de dado requisitado

15

## Bloqueio Exclusivo e Compartilhado

- se bloqueio ( *item\_dado* ) = *read\_locked*  
então *item\_dado* pode ser acessado desde que a operação do BD que o requisitou seja de leitura
- se bloqueio ( *item\_dado* ) = *write\_locked*  
então *item\_dado* não pode ser acessado
- se bloqueio ( *item\_dado* ) = *unlocked*  
então *item\_dado* pode ser acessado

16

## Bloqueio Exclusivo e Compartilhado

- Operações adicionais
  - *read\_lock ( item\_dado )*
  - *write\_lock ( item\_dado )*
  - *unlock ( item\_dado )*
- Operações *read\_lock*, *write\_lock* e *unlock*
  - são unidades indivisíveis
  - utilizam conceitos semelhantes ao de região crítica em Sistemas Operacionais

17

## Regras Seguidas pelas Transações

- uma transação T deve realizar a operação *read\_lock(x)* ou *write\_lock(x)* antes de qualquer operação *read\_item(x)* de T
- uma transação T deve realizar a operação *write\_lock(x)* antes de qualquer operação *write\_item(x)* de T
- uma transação T deve realizar a operação *unlock(x)* depois de todas as operações *read\_item(x)* ou *write\_item(x)* de T

18

## Regras Seguidas pelas Transações

- Uma transação T não realizará uma operação *read\_lock(x)* se ela já possui um bloqueio compartilhado ou exclusivo de x
- uma transação T não realizará uma operação *write\_lock(x)* se ela já possui um bloqueio compartilhado ou exclusivo de x
- uma transação T não realizará uma operação *unlock(x)* a menos que ela possua um bloqueio compartilhado ou exclusivo de x

19

## Problema

- Bloqueios binários ou de modo múltiplo
  - não garantem a serialização dos escalonamentos nos quais as transações participam

20

## Exemplo

- | Transação 1    | Transação 2    |
|----------------|----------------|
| read_lock (y)  | read_lock (x)  |
| read_item (y)  | read_item (x)  |
| unlock (y)     | unlock (x)     |
| write_lock (x) | write_lock (y) |
| x := x + y     | y := x + y     |
| write_item (x) | write_item (y) |
| unlock (x)     | unlock (y)     |

21

Transação 1	Transação 2
read_lock (y) read_item (y) unlock (y)	read_lock (x) read_item (x) unlock (x) write_lock (y) read_item (y) y := x + y write_item (y) unlock (y)
write_lock (x) read_item (x) x := x + y write_item (x) unlock (x)	

Execução  
concorrente  
não  
serializável

22

## Conversão de Bloqueios

- As regras apresentadas na página 18 podem ser relaxadas, como a seguir:
- upgrade (ampliação): conversão de bloqueio compartilhado para bloqueio exclusivo
  - *read\_lock(item\_dado)* → *write\_lock (item\_dado)*
- downgrade (redução): conversão do bloqueio exclusivo para bloqueio compartilhado
  - *write\_lock(item\_dado)* → *read\_lock(item\_dado)*

23