

# Processamento de Transações e Recuperação a Falhas em BDs

Disciplina Bancos de Dados II

Prof. Renato Fileto  
INE/CTC/UFSC

## Tópicos

1. **Introdução**
  - Conceituação e a necessidade do controle de sua execução
  - Operações de transações
  - Estados do ciclo de vida e propriedades desejáveis de transações
  - Backup, log e escalonamentos
2. **Caracterização de escalonamentos quanto a recuperabilidade, ausência de *rollback* em cascata e rigor**
3. **Caracterização de escalonamentos quanto à serialidade**
  - Algoritmo para a detecção de conflitos de serialidade
4. **Técnicas de controle de transações**
  - Trancas e *2-Phase Lock (2PL)*
5. **Técnicas de recuperação de banco de dados**
6. **Bibliografia, leituras recomendadas e listas de exercícios**

## Transação

**Transação** é a "execução de um trecho de programa" que **acessa** ou **modifica** o conteúdo da base de dados.

[Elmasri & Navathe, 1996]

### Exemplo 1

**T1**  
read\_item (X);  
X:=X-N;  
write\_item (X);  
read\_item (Y);  
Y:=Y+N;  
write\_item (Y);

### Exemplo 2

**T2**  
read\_item (X);  
X:=X+M;  
write\_item (X);

## Operações de Leitura e de Escrita

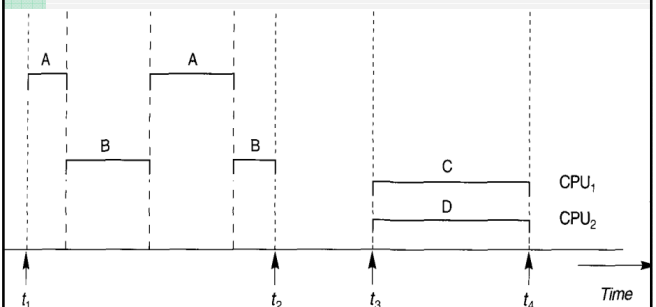
- **read\_item(X) ou R(X)**: lê item de dado X para variável de programa
  1. Encontra endereço do bloco de disco contendo X
  2. Copia o bloco do disco no buffer de memória (**se não estiver previamente no buffer**)
  3. Copia item X do buffer para variável de programa
- **write\_item(X) ou W(X)**: escreve item X na base de dados
  1. Encontra endereço do bloco de disco contendo X
  2. Copia o bloco no buffer de memória (se não estiver)
  3. Copia item X da variável de programa para buffer
  4. Grava bloco do buffer no disco (**imediatamente ou mais tarde**)

## Problemas

- SGBDs multi-usuários
- Sistemas multi-processo e programação concorrente:
  - tempo compartilhado (execução intercalada)
  - múltiplos processadores (execução simultânea)
- Manutenção da integridade da base de dados
- Falhas de *Hardware* e *Software*
  - Necessidade de controle da execução de transações
  - Necessidade de mecanismos para recuperação do BD

## Sistemas de Tempo Compartilhado e Sistemas Multiprocessados

Elmasri/Navathe 2009



## Possíveis problemas decorrentes da falta de controle do proc. trans.

- Perda de dados atualizados por transações
- Execução parcial de uma transação
- Problemas de integridade
  - escrita perdida (*lost write*)
  - atualização temporária ou leitura suja (*dirty read*)
  - soma incorreta (*incorrect summary*)
  - :

## Escrita perdida (*lost write*)

	Transação 1	Transação 2
tempo ↓	<code>read_item(X);</code>	<code>read_item(X);</code>
	<code>X:=X-N;</code> <b><code>write_item(X);</code></b>	<code>X:=X-M;</code> <b><code>write_item(X);</code></b>

*A subtração de N é perdida!*

## Atualização temporária (*temporary update*) ou leitura suja (*dirty read*)

	Transação 1	Transação 2
tempo ↓	<code>read_item(X);</code> <code>X:=X-N;</code> <code>write_item(X);</code>	<code>read_item(X);</code> <code>X:=X-M;</code> <code>write_item(X);</code>

**FALHA!!!**

*Operações de ambas as transações precisarão ser desfeitas!*

## Soma incorreta (*temporary update*) ou leitura suja (*dirty read*)

	Transação 1	Transação 2
tempo ↓	<code>read_item(X);</code> <code>X:=X*1.1;</code> <code>write_item(X);</code>	<code>Sum := 0;</code>
	<code>read_item(Y);</code> <code>Y:=Y*1.1;</code> <code>write_item(Y);</code>	<b><code>read_item(X);</code></b> <b><code>Sum += X;</code></b> <b><code>read_item(Y);</code></b> <b><code>Sum += Y;</code></b>

*X foi somado atualizado mas Y não!!!*

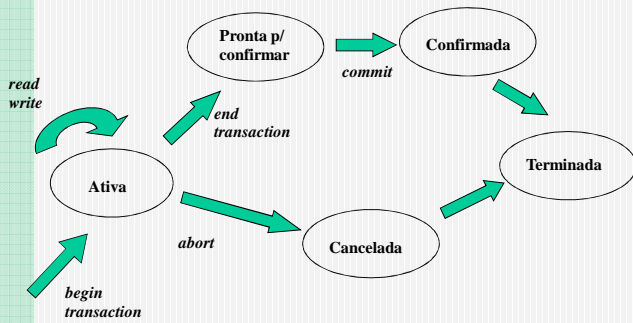
## Tipos de falhas

- Falha do sistema computacional (hardware, software, rede)
- Falha na transação
- Erros locais ou exceções
- Falhas devido ao controle de concorrência
- Falhas de disco
- Problemas físicos
- Catástrofes

## Operações adicionais de transações

- BEGIN\_TRANSACTION
- READ / WRITE
- END\_TRANSACTION
- COMMIT\_TRANSACTION
- ROLLBACK (ABORT)

## Estados de uma transação



## Marcas do LOG do sistema de BDs

[start-transaction,T]

[write\_item,T,X,old\_value,new\_value]

[read\_item,T,X]

[commit,T]

[abort,T]

[checkpoint]

## Propriedades "ACID" das transações

- **Atomicidade:** todas as operações de uma transação são realizadas ou nenhuma delas
- **Consistência:** leva o BD de um estado consistente para outro estado consistente
- **Independência/Isolamento:** o processamento de uma transação não deve interferir em outras, sendo o resultado do processamento concorrente equivalente ao do processamento serial
- **Durabilidade (ou permanência):** resultados de operações confirmadas não podem ser perdidos

## Caracterização de escalonamentos quanto à recuperabilidade

- **Recuperável:** nenhuma transação T que tenha lido item alterado por outra transação T' confirma antes de T'  
**Exemplo de violação:** R1(X)W1(X)R2(X)W2(X)C2
- **Livre de Rollback em Cascata:** nenhuma transação lê item previamente escrito por outra que não tenha confirmado  
**Exemplo de violação:** R1(X)W1(X)R2(X)W2(X)
- **Estrito:** nenhuma transação lê nem escreve item previamente escrito por outra que não tenha confirmado  
**Exemplo de violação:** R1(X)W1(X)W2(X)

## Caracterização de escalonamentos quanto à serialidade

- **Serial:** **não** há intercalação de operações de diferentes transações
- **Não-serial:** há intercalação de operações de diferentes transações
- **Seriável:** a execução equivale à de um escalonamento serial
- **Não-Seriável:** a execução **não** equivale à de um escalonamento serial

## Operações conflitantes

- Par de operações de um mesmo escalonamento tal que ao menos uma delas é *WRITE* e a outra *READ* ou *WRITE*
- Realizadas por transações distintas
- Atuando sobre um mesmo item de dados

**Exemplo:** R2(Y)R1(X)W1(X)R1(Y)R2(X)

### Importante:

- Um escalonamento não é seriável se seu grafo de precedência de operações conflitantes apresentar qualquer ciclo.