
Banco de Dados I

6 – Transações e Controle de Concorrência

Grinaldo Lopes de Oliveira (grinaldo@gmail.com)
Curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas

Agenda

- **Aprendendo**
 - **Transações**
 - **Controle de Concorrência**





Transação

O que é uma Transação

- O termo transação refere-se a uma coleção de operações que forma uma única unidade lógica de trabalho.
 - Por exemplo, uma transferência de dinheiro de uma conta para outra é uma transação consistindo de duas atualizações, uma para cada conta.
-

Exemplo de Transações

T1: Transação T₁
A=ler_item (X);
A = A - N;
escrever_item (X,A);
B=ler_item (Y);
B = B + N;
escrever_item (Y,B);

T2: Transação T₂
C=ler_item (X);
C = C + M;
escrever_item (X,C);

Transações de vários
usuários



- (1) executadas
concorrentemente
- (2) podem acessar e atualizar o
mesmo item de dados

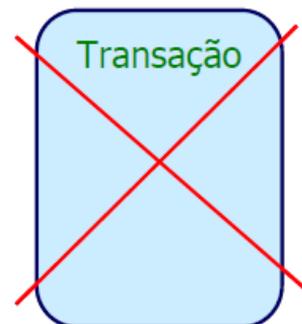
Transações do SGBD

Na execução de uma **transação** o **SGBD** deve garantir:

(1) **Todas as operações** na transação foram completadas com **sucesso** e seu efeito será **gravado permanentemente** no BD

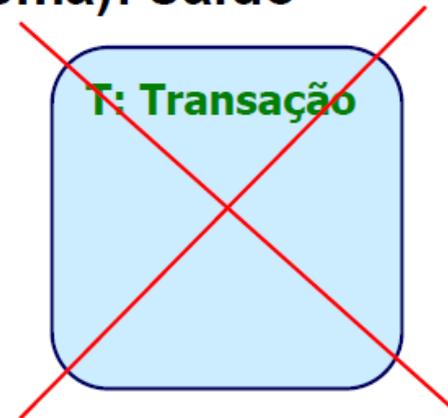


(2) Transação **NÃO** terá nenhum **efeito sobre o BD** ou outras transações (**TRANSAÇÃO FALHAR DURANTE EXECUÇÃO**)

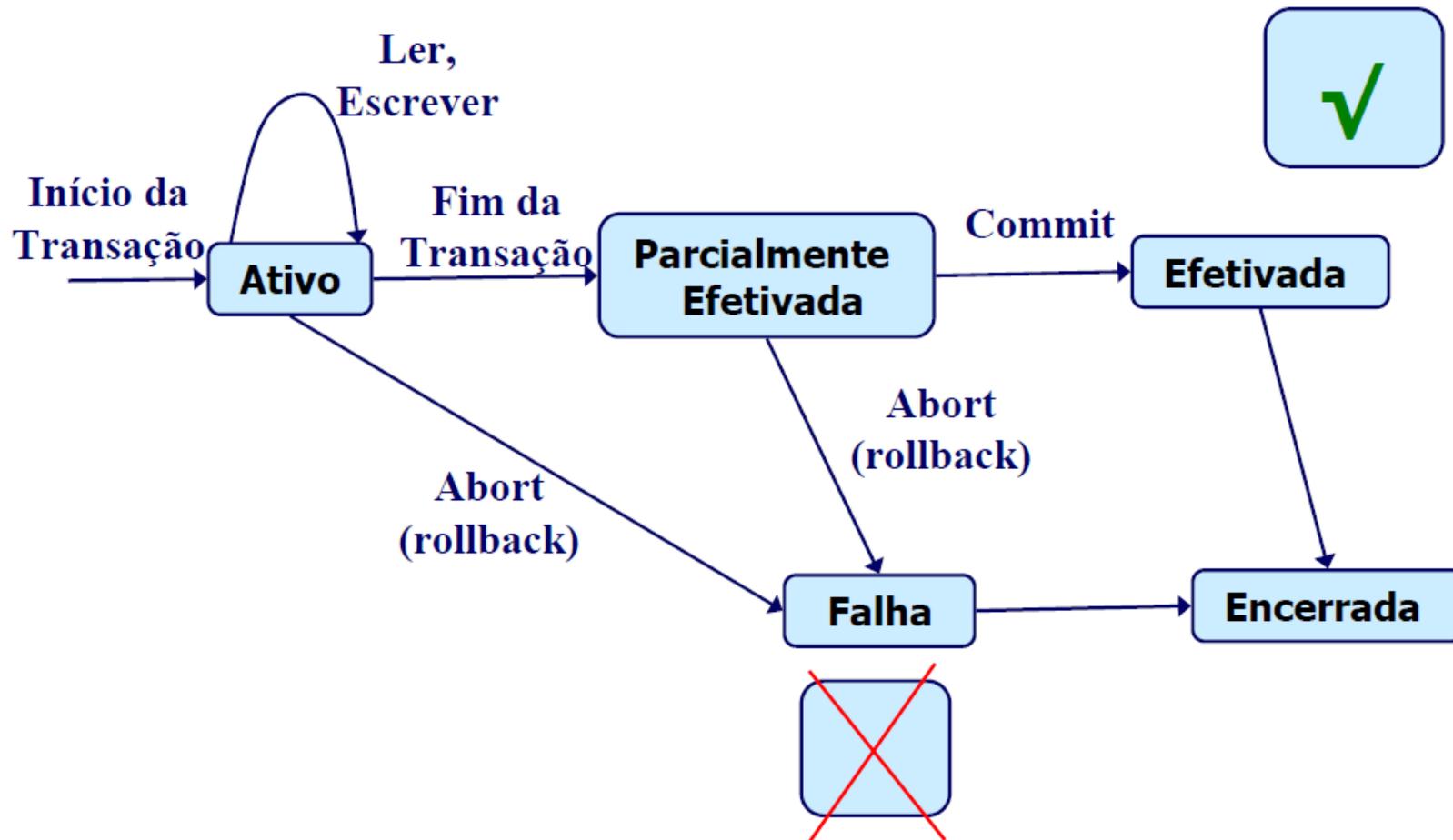


O que faz uma transação falhar?

- ◆ **Computador falhar por hardware, software ou rede**
- ◆ **Erro durante execução de operação na transação: estouro de variáveis**
- ◆ **Condições de exceção detectadas pela transação (necessitam o cancelamento da mesma): saldo insuficiente em conta**
- ◆ **Falta de energia, ar-condicionado**



Estados de uma Transação



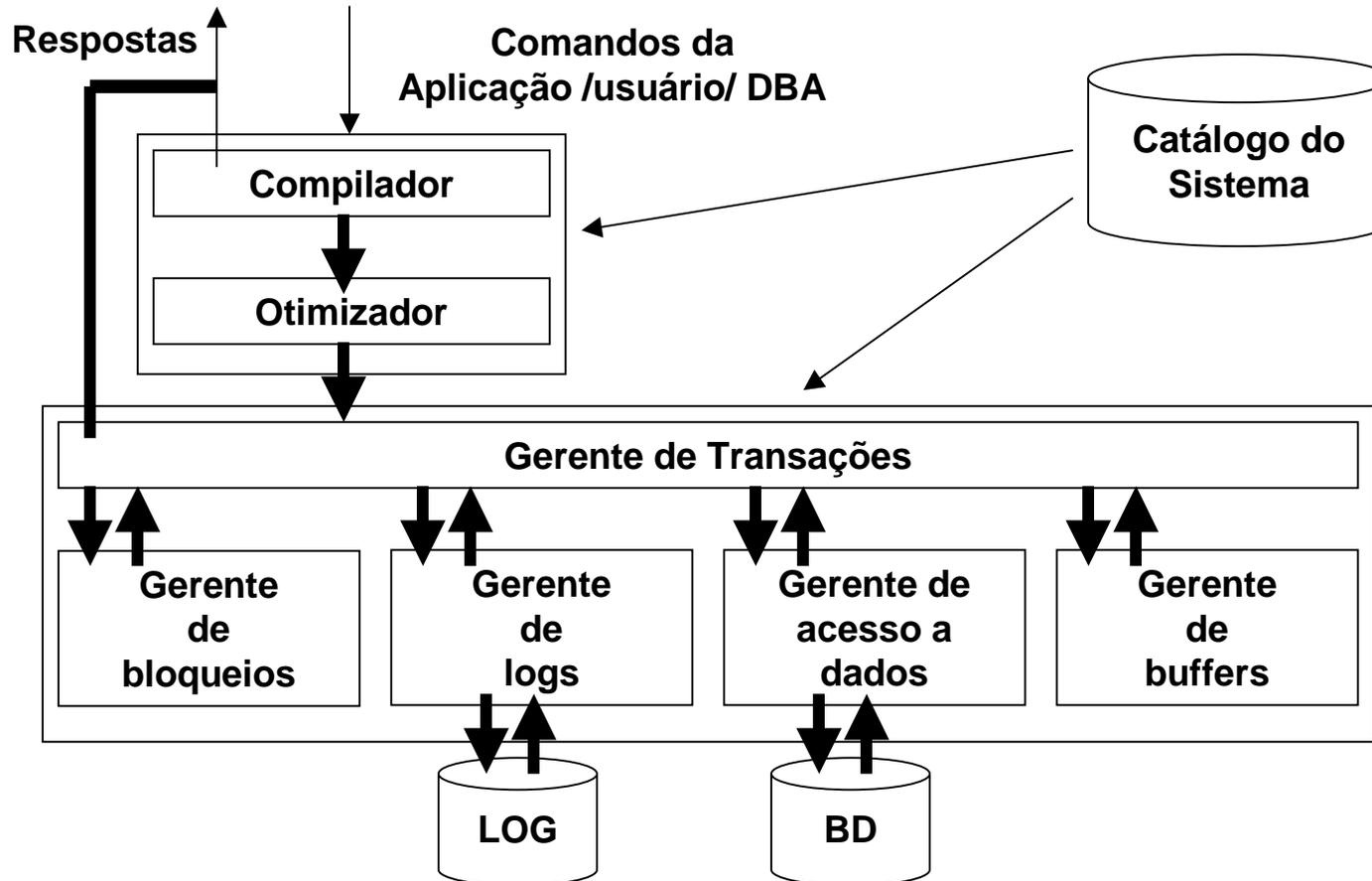
Propriedades de uma Transação

■ ACID

- ❑ A: Atomicidade
 - Unidade lógica atômica (tudo ou nada)
 - ❑ C: Consistência (Preservação)
 - Ao final de uma transação o banco continua consistente
 - ❑ I: Isolamento
 - A execução de uma transação não deve sofrer interferência de outras transações concorrentes
 - ❑ D: Durabilidade (Persistência)
 - Após o ponto de confirmação, as alterações devem persistir no banco de dados
-

Arquitetura de um SGBD

Referência





Controle de Concorrência

Controle de Concorrência

Primitiva

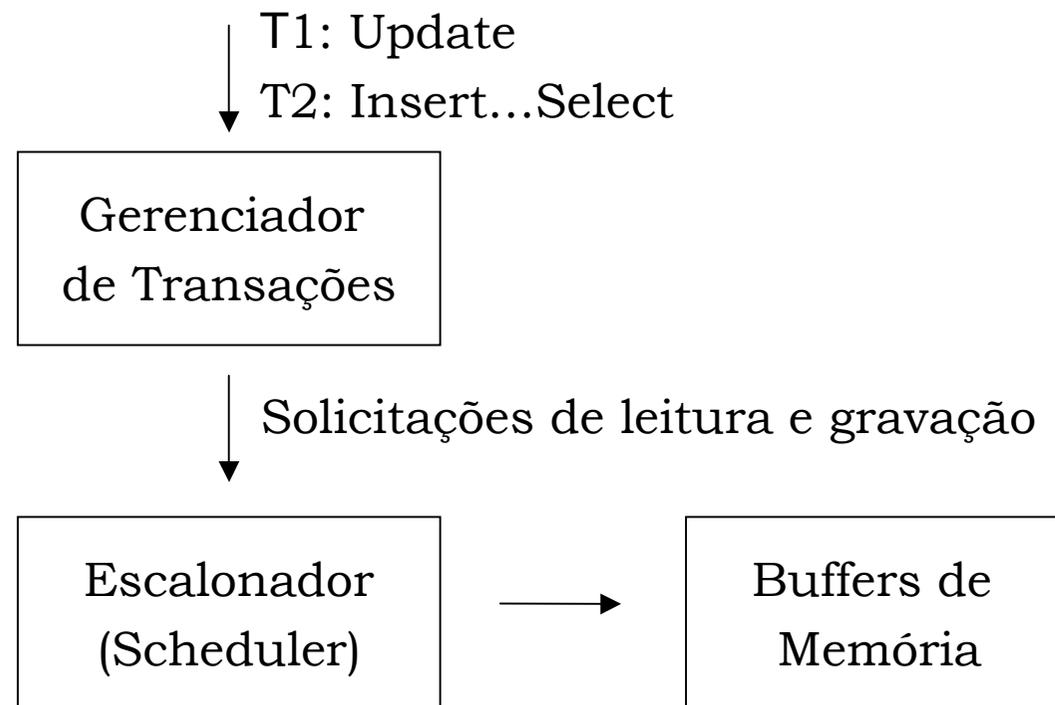
“Princípio da correção”

- Toda transação, se executada de forma isolada, transformará qualquer estado consistente em outro estado consistente.
-

Controle de Concorrência

Escalonador (Scheduler)

- Escalonador : Tem como função assegurar que as transações preservem a consistência quando executadas simultaneamente.



Controle de Concorrência

Escalonamento

- Seqüência ordenada das ações executadas por uma ou mais transações.

Ex: Restrição: $A=B$

Inicialmente possuem valor = 25

Ex.: **T1**

```
Read(A,t)
T := t + 100;
Write(A,t)
```

```
Read(B,t)
T := t + 100;
Write(B,t);
```

T2

```
Read(A,s)
s := s*2;
Write(A,s)
```

```
Read(B,s)
s:= s * 2;
Write(B,s);
```

Controle de Concorrência

Escalonamento

Escalonamento Serial (1) – T1 antes de T2

Ex1:

T1

Read(A,t)

t := t + 100;

Write(A,t)

Read(B,t)

t := t + 100;

Write(B,t);

T2

Read(A,s)

s := s*2;

Write(A,s)

Read(B,s)

s:= s * 2;

Write(B,s);



Controle de Concorrência

Escalonamento

Escalonamento Serial (2) – T2 antes de T1

Ex2:

T2
Read(A,s)
s := s*2;
Write(A,s)

Read(B,s)
s:= s * 2;
Write(B,s);

T1
Read(A,t)
t := t + 100;
Write(A,t)

Read(B,t)
t := t + 100;
Write(B,t);



Controle de Concorrência

Escalonamento

- Serialização (Serializabilidade, Seriabilização)
 - Requisito abstrato que tem como objetivo assegurar que as transações executem de forma concorrente e ao final o estado do banco continue consistente.
 - Escalonamento Serializável:
 - Quando seu efeito sobre o estado do banco de dados é igual ao de algum escalonamento serial.
 - Escalonamento não-serializável:
 - Escalonamento que não assegura o estado consistente do banco de dados
-

Controle de Concorrência

Escalonamento

- Escalonamento Serializável

Ex3:

T1

Read(A,t)
t := t + 100;
Write(A,t)

Read(B,t)
t := t + 100;
Write(B,t)

T2

Read(A,s)
s := s *2;
Write(A,s);

Read(B,s)
s := s *2;
Write(B,s);



Controle de Concorrência

Escalonamento

Escalonamento Não-Serializável

Ex4:

T1

Read(A,t)
t := t + 100;
Write(A,t)

T2

Read(A,s)
S := s *2;
Write(A,s);

Read(B,s)
S := s *2;
Write(B,s);

Read(B,t)
t := t + 100;
Write(B,t)



Controle de Concorrência

Serialização

- **Serializabilidade de conflito**

Baseia-se na idéia de conflito de operações. Ordenamento de pares de operações conflitantes:

Operações conflitantes: (Leitura e gravação) ou (gravação e gravação) de um mesmo item por transações diferentes.

R1(X) e W2(X);

W1(X) e R2(X);

W1(X) e W2(X);

Controle de Concorrência

Bloqueios

- Como os SGBDs impõem a serialização ?
 - A técnica mais comum adotada é a utilização de bloqueios sobre elementos do banco de dados, a fim de evitar um comportamento não serializável. (Pessimista)
 - Uma transação obtém bloqueios sobre os elementos do banco de dados para impedir que outras transações acessem esses elementos ao mesmo tempo e, portanto, haja risco da não serialização.
 - É utilizada uma tabela de bloqueios
 - Há outras técnicas:
 - Timbre de hora – Timestamp
 - Técnicas otimistas- Entendem que operações conflitantes são exceção
-

Controle de Concorrência

Bloqueios

- Regras
 - Uma transação só pode ler ou gravar um elemento se tiver solicitado previamente um bloqueio sobre esse elemento.
 - $Li(x)$ - Lock
 - Se uma transação bloquear um elemento, ele terá de desbloquear mais tarde
 - $Ui(x)$ - Unlock
-

Controle de Concorrência

Bloqueios

- Tabela de bloqueio
 - De uma maneira geral, é uma relação que associa elementos do banco de dados com informações de bloqueio sobre esses elementos.

A	Modo de grupo: U, Esperando: sim, Lista: T2-X, T3-S, T4-U
B	...
F

Controle de Concorrência

Bloqueios

- Tipos de bloqueios
 - ❑ Compartilhado (leitura-shared) - S
 - ❑ Exclusivo (gravação-exclusive) - X
 - ❑ Atualização (upgrade) - U

	S	X	U
S	Sim	Não	Sim
X	Não	Não	Não
U	Sim	Não	Não

Controle de Concorrência

Bloqueios

- Granularidade dos bloqueios
 - Linha (registros)
 - Página de disco
 - Tabela
 - Banco de Dados
 - O SGBD decide automaticamente o melhor nível de granularidade (Bloqueio de múltipla granularidade).
 - Alguns SGBDs permitem que o usuário altere a granularidade e o tipo do bloqueio.
-

Controle de Concorrência

Escalonamento (bloqueio)

- Escalonamento válido por bloqueio mas Não-Serializável.

Ex:

T1	T2
XL1; R1(A)	
A := A + 100;	
W1(A);U1(A)	
	XL2(A); R2(A)
	A := A *2;
	W2(A); U2(A)
	XL2(B); R2(B)
	B := B *2;
	W2(B); U2(B);
XL1(B); R1(B)	
B := B + 100;	
W1(B); U1(B);	



- O escalonamento acima é válido porque as duas transações não contêm um bloqueio sobre A e nem B ao mesmo tempo. Mas o mesmo não é serializável.
- Percebemos que um bloqueio de uma fase não é garantia de serializabilidade.

Controle de Concorrência

Bloqueios 2PL

- Bloqueios 2PL (two-phase locking)
 - Forma de garantir a seriabilidade
 - Também chamado de bloqueio em duas fases
 - Em toda transação, todas as solicitações de bloqueio precedem todas as solicitações de desbloqueio
 - Primeira Fase: todos os bloqueios são obtidos pela transação e nenhum bloqueio é liberado (fase de crescimento)
 - Segunda Fase: Os bloqueios são liberados mas nenhum é obtido (Fase de encolhimento)
 - Problema não resolvido pelo 2PL: impasses (deadlock)
-

Controle de Concorrência

Bloqueios em 2PL

- Escalonamento válido por bloqueio e Serializável.

Exemplo:

T1

XL1(A); R1(A)
A := A + 100;
W1(A);XL1(B);U1(A)

R1(B)
B := B + 100;
W1(B); U1(B);

T2

XL2(A); R2(A)
A := A *2;
W2(A); XL2(B); (Negado)

XL2(B);U2(A)
R2(B)
B := B *2;
W2(B); U2(B);



Controle de Concorrência

Deadlock

- Ocorre quando uma transação T1 está esperando a liberação de algum bloqueio realizado por T2 e vice-versa.
 - Protocolos de prevenção de deadlock
 - Geralmente não é muito viável
 - (bloqueios antecipados ou ordenação igual na operações no banco de dados)
 - Protocolos de detecção de deadlock
 - Mais viável
 - Uso de timeouts
 - o SGBD procura abortar a transação que atualizou menos dados ou que esteja rodando a menos tempo.

Exemplo:

T1

R1(A)

W1(B)

T2

R2(B)

W2(A)

DEADLOCK

Controle de Concorrência

Problemas de Concorrência

- Problemas clássicos de concorrência
 - Atualização perdida (Lost Update)
 - Leitura suja (Dirty read)
 - Leitura não-repetitiva (Repeatable-read)
 - Fantasmas (Phantoms)
-

Controle de Concorrência

Problemas de Concorrência

- Atualização perdida (Lost Update)
 - Ocorre quando duas transações que acessam os mesmos itens do banco de dados têm suas operações entrelaçadas, de modo que torne incorreto o valor de algum item.
 - Leitura Suja (Dirty read)
 - Ocorre quando uma transação atualiza um item do banco de dados e, por algum motivo, essa transação falha, sendo que outra transação leu esse item supostamente atualizado
-

Controle de Concorrência

Problemas de Concorrência

- Leitura Não Repetitiva (Non-Repeatable read)
 - Ocorre quando uma transação lê itens em uma determinada condição e depois outra transação altera e efetiva um novo item que satisfaz a condição da transação anterior. Caso a primeira transação faça novamente uma leitura com a mesma condição, os registros podem aparecer de maneira diferente.
 - Fantasmas (Phantoms)
 - Ocorre quando uma transação lê itens em uma determinada condição e depois outra transação insere e efetiva um novo item que satisfaz a condição da transação anterior. Caso a primeira transação faça novamente uma leitura com a mesma condição, um registro “fantasma” irá aparecer
-

Controle de Concorrência

Níveis de Isolamento (Padrão SQL-92)

- Read uncommitted
 - Permite a leitura de dados não efetivados
 - Há ganho de performance, mas perda de segurança.
 - Read committed
 - Nível de isolamento Padrão
 - Leitura só pode ser feita em dados efetivados.
 - Repeatable Read
 - Registros lidos não serão alterados por outros processos, garantindo releituras idênticas.
 - Serializable
 - A mais restrita de todas.
 - Inserções ou deleções não podem ser feitas em conjuntos de registros lidos.
-

Controle de Concorrência

Níveis de Isolamento (Padrão SQL-92)

- Sintaxe SQL
 - A configuração do nível de isolamento pode ocorrer para todas as transações do banco de dados ou para alguma específica.
- SET TRANSACTION
 - ISOLATION LEVEL
 - [
READ UNCOMMITTED ou
READ COMMITTED *(Padrão)
REPEATABLE READ
SERIALIZABLE
]



Debate em Sala de Aula

- Uma aplicação está com baixo tempo de resposta em virtude de muitas transações concorrentes e alto grau de bloqueios. O que você faria para minimizar este problema?



Banco de Dados I

6 – Transações e Controle de Concorrência

Grinaldo Lopes de Oliveira (grinaldo@gmail.com)
Curso Superior de Tecnologia em
Análise e Desenvolvimento de Sistemas