

SISTEMAS DE BANCO DE DADOS 1

AULA 12

Transação em Banco de Dados

Vandor Roberto Vilardi Rissoli



APRESENTAÇÃO

- Conceitos e Definições sobre Transação em Banco de Dados
- Transação e Concorrência
- Algumas situações de conflito com a Concorrência
- Referências



TRANSAÇÃO

As operações que formam uma única unidade lógica de trabalho são chamadas de TRANSAÇÕES.

Exemplo:

A transferência de um valor de uma conta para outra conta, na visão do cliente, consiste de uma operação única e simples. No BD porém, ela envolve várias operações para que esta transferência seja executada com sucesso.

Suponha que o valor seja debitado na conta a ser retirada este valor, mas por uma falha este mesmo valor não tenha sido creditado na outra conta.



TRANSAÇÃO

É uma unidade de execução de programa que acessa e manipula dados no BD. Geralmente ela consiste na execução de um programa (instruções) elaborado com:

- linguagem de manipulação de dados (alto nível);
- linguagem de programação;

→ A **transação** consiste em todas as operações a serem executadas a partir do começo até o fim da transação.



TRANSAÇÃO

Para assegurar a integridade dos dados, um BD deve garantir sempre algumas propriedades das transações. Estas propriedades são:

Atomicidade: Todas as operações da transação são refletidas corretamente no BD ou nenhuma será

Consistência: A execução de uma transação isolada preserva a consistência do BD (situação inicial e final)

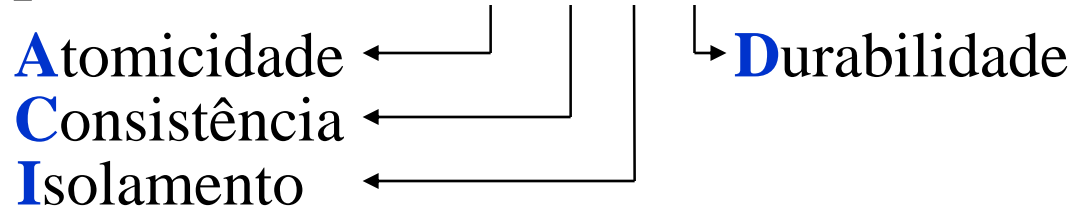
Isolamento: Cada transação não toma conhecimento das outras transações concorrentes

Durabilidade: Depois da transação completar-se com sucesso, as mudanças que ela faz no BD, persistem até mesmo se houver falhas no sistema



TRANSAÇÃO

Estas propriedades também são conhecidas pelo acrônimo **A C I D**



Exemplo:

Suponha um sistema bancário simplificado com várias contas e diversas transações sobre estas contas. A transação (**T**) de transferência de cem reais (R\$ 100,00) de uma conta **A** para uma outra conta **B** seria representada por:

T: leia(A);
A = A - 50;
escreva(A);
leia(B);
B = B + 50;
escreva(B);

TRANSAÇÃO

Suponha neste exemplo também que a operação **escreva** consista na gravação, do dado presente na memória para o disco magnético, enquanto que a operação **leia** seja a transferência dos dados desejados do disco para a memória.

CONSISTÊNCIA: sendo o BD consistente antes da operação, ele deve continuar consistente após a transação.

Para A valendo R\$ 400,00 e B valendo R\$ 200,00 tem-se:

T: leia(A);

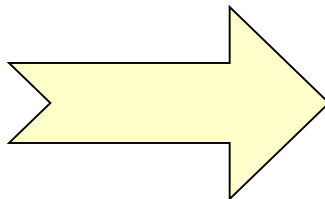
$A = A - 50;$

escreva(A);

leia(B);

$B = B + 50;$

escreva(B);



$A=400$

$A=400 - 50$

$A=350$ (gravar em disco)

$B=200$

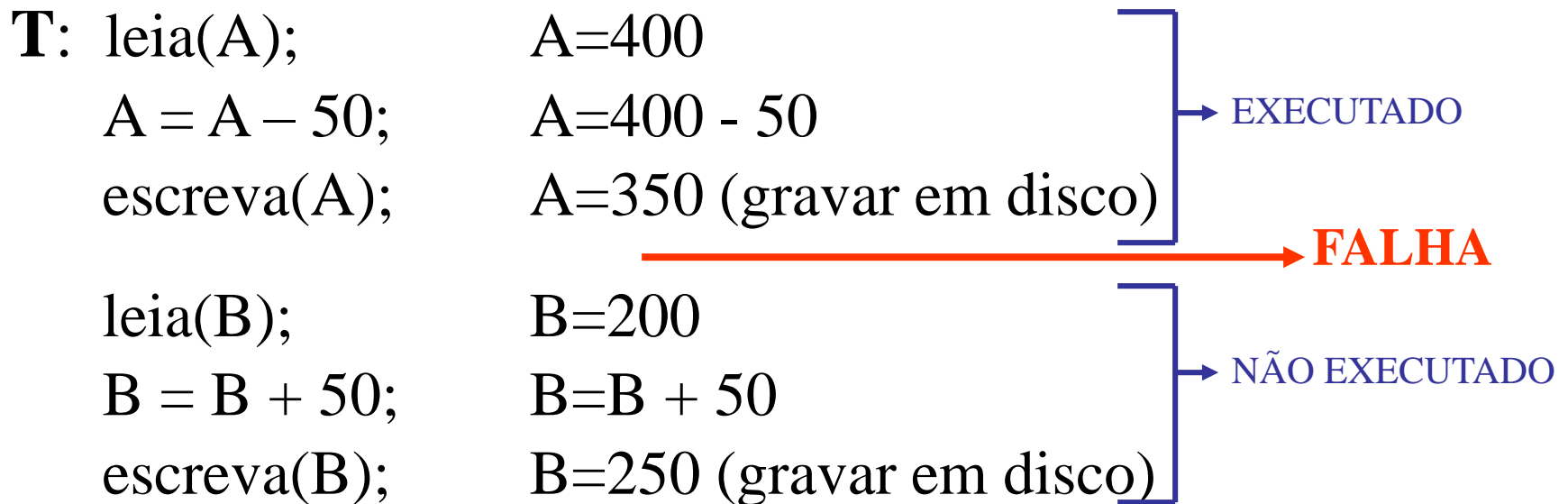
$B=200 + 50$

$B=250$ (gravar em disco)

TRANSAÇÃO

ATOMICIDADE: ou a transação é executada por completo ou nenhuma de suas partes serão.

Seguindo o mesmo exemplo tem-se:



→ Desfaz-se todas as operações desta transação de transferência de dados, ou ela é executada totalmente.

TRANSAÇÃO

DURABILIDADE: completada a transação com sucesso, todas as atualizações realizadas no BD persistirão, até mesmo se houver uma falha de sistema após a transação ser completada.

Seguindo no exemplo bancário, se a transação for completada com sucesso:

- notificação confirmando a operação para o usuário;
- não ocorreu nenhuma falha de sistema;
- confirmada a transferência que persistirá no BD.

→ **Assegurar esta propriedade** é responsabilidade de um componente do sistema de BD denominado

GERENCIAMENTO DE RECUPERAÇÃO



TRANSAÇÃO

ISOLAMENTO: garante que a execução concorrente de transações, **resulte em uma situação equivalente no sistema ao resultado** obtido pela **realização das transações uma de cada vez (serial)**, em qualquer ordem.

Seguindo no exemplo bancário, é possível observar que o BD fica temporariamente inconsistente, pois enquanto a transação transfere fundos de **A** para **B**, existe um momento em que se deduziu o valor de **A**, mas ainda não foi acrescentado a **B**.

T: leia(A);	A=400	
A = A - 50;	A=400 - 50	
escreva(A);	A=350 (gravar em disco)] → Período de inconsistência do BD
leia(B);	B=200	
B = B + 50;	B=B + 50	
escreva(B);	B=250 (gravar em disco)	

TRANSAÇÃO

→ Assegurar a propriedade de ISOLAMENTO é responsabilidade de um componente do sistema de BD denominado **CONTROLE DE CONCORRÊNCIA**.

ESTADO DA TRANSAÇÃO

Para melhor compreensão dos possíveis estados de uma transação será usado um **modelo simples e abstrato** destas situações:

ATIVA: permanece neste estado enquanto está sendo executada;

EFETIVAÇÃO PARCIAL: após execução da última declaração;

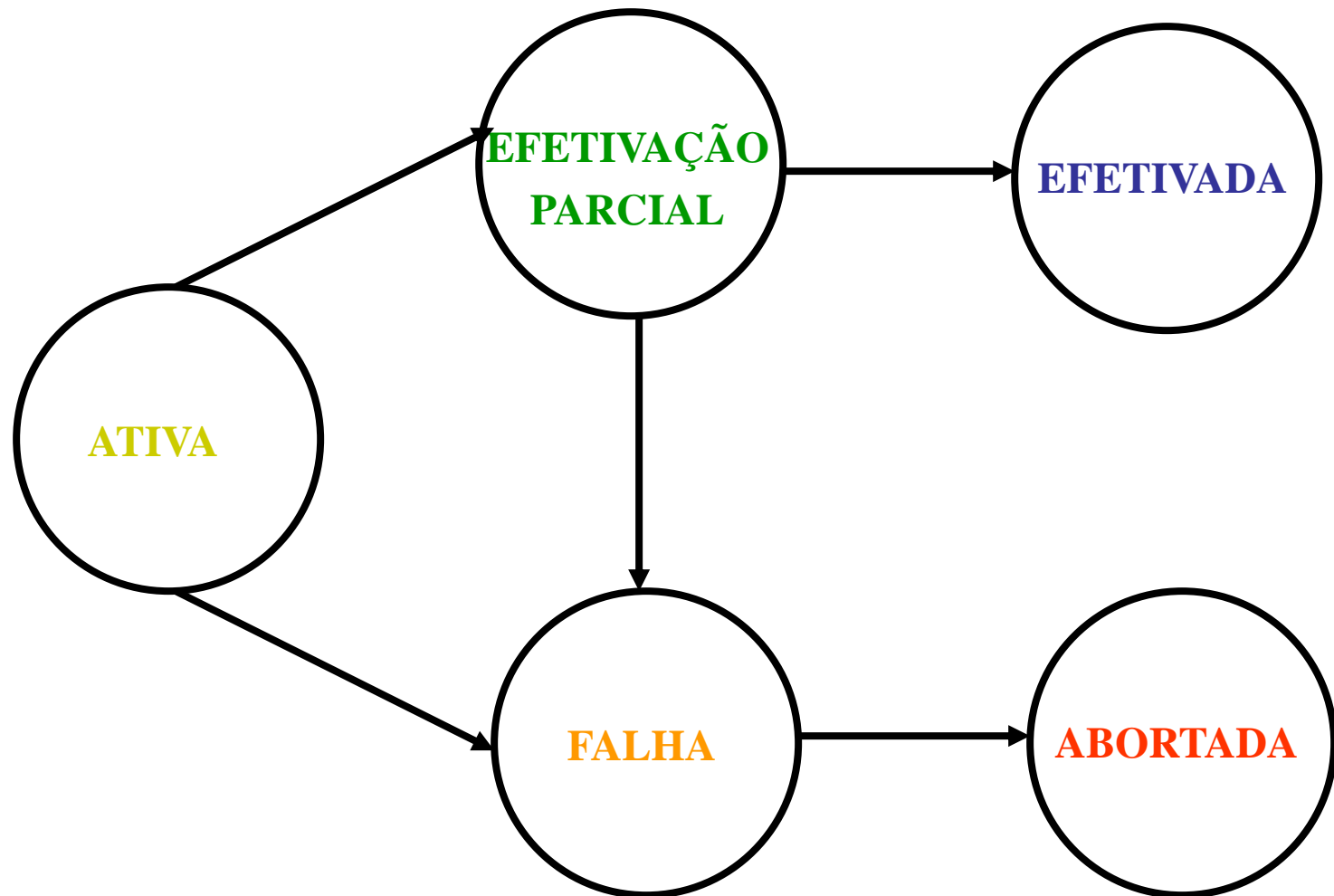
FALHA: descobre-se que a execução não poderá ser efetivada;

ABORTADA: transação desfeita, restabelecendo o BD ao início;

EFETIVADA: após a conclusão com sucesso;

TRANSAÇÃO

DIAGRAMA DE ESTADO



TRANSAÇÃO

- Diz-se que uma transação foi **efetivada** (*committed*) somente se ela entrar no estado de **EFETIVADA**;
 - Diz-se que uma transação foi **abortada** (*rolled back*) somente se ela entrar no estado de **ABORTADA**;
 - Uma transação é concluída se estiver no estado de **EFETIVADA** ou de **ABORTADA**.
- Uma transação de compensação pode desfazer os efeitos de uma transação efetivada, porém nem sempre isso é possível. Esta transação tem a responsabilidade de criação e execução deixada a cargo do usuário.

TRANSAÇÃO

Uma transação no estado de falha (**erro** de hardware, ou lógico, ou de leitura, entre outros) **não pode prosseguir** com sua execução normal, devendo ser desfeita.

Assim, ela passa para o estado de **abortada**, onde pode:

- **Reiniciar a transação**: possível somente para erros de hardware ou software e **não pela lógica da operação**;
- Encerrar a transação: **erro lógico** normalmente, pois a aplicação (ou programa) deverá ser refeito;

→ A operação que reinicia uma transação consiste na criação de uma nova transação para ser processada.



TRANSAÇÃO

INTRODUÇÃO A TRANSAÇÕES CONCORRENTES

O processamento de transações concorrentes agiliza a realização da tarefa desejada, mas também traz diversas complicações em relação a consistência dos dados no BD.

Seria muito mais fácil manter as execuções das transações **sequencialmente**, mas duas possibilidades básicas incentivam a concorrência e sua agilização:

- Operação da CPU e as E/S podem ser feitos em paralelo;
- Mistura de transações simultâneas no sistema
 - curtas;
 - longas;

→Acessa diferentes partes do BD;

→ Reduz atrasos imprevisíveis;

→ Diminui o tempo médio de resposta;

→Reduz ociosidade da CPU, discos e outros dispositivos.

TRANSAÇÃO

O processamento concorrente compromete a propriedade de consistência do BD.

Para permitir a concorrência eficiente, sem comprometer a consistência, é analisada a escala de execução (*schedules*) das transações envolvidas.

Exemplo:

Continuando com o sistema bancário, tendo diversas contas com vários lançamentos que acessam e atualizam estas contas, supõem-se duas transações T_1 e T_2 que transferem fundos de uma conta **A** para outra conta **B**.

→ *throughput* do sistema – quantidade de transações que podem ser executadas em um determinado tempo.

TRANSAÇÃO

T₁: leia(A);	A=400
A = A - 50;	A=400 - 50
escreva(A);	A=350 (gravar em disco)
leia(B);	B=200
B = B + 50;	B=B + 50
escreva(B);	B=250 (gravar em disco)

A outra transação transfere 10% da conta **A** para a conta **B**.

T₂: leia(A);	A=350
aux = A * 0.10;	aux=350 * 0.10
A = A - aux;	A=350 - 35
escreva(A);	A=315 (gravar em disco)
leia(B);	B=250
B = B + aux;	B=B + 35
escreva(B);	B=285 (gravar em disco)

TRANSAÇÃO

T ₁	T ₂
leia(A); A = A - 50; escreva(A); leia(B); B = B + 50; escreva(B);	leia(A); aux = A * 0.10; A = A - aux; escreva(A); leia(B); B = B + aux; escreva(B);
<div>ESCALA 1</div>	<div>ESCALA 2</div>
<hr style="border-top: 1px dashed black;"/> A=350 B=250	<hr style="border-top: 1px dashed black;"/> A=315 B=285

→ 600

→ 600

T ₁	T ₂
leia(A); aux = A * 0.10; A = A - aux; escreva(A); leia(B); B = B + aux; escreva(B);	leia(A); aux = A * 0.10; A = A - aux; escreva(A); leia(B); B = B + aux; escreva(B);
<div>ESCALA 1</div>	<div>ESCALA 2</div>
<hr style="border-top: 1px dashed black;"/> A=310 B=290	<hr style="border-top: 1px dashed black;"/> A=360 B=240

→ 600

→ 600

TRANSAÇÃO

- Estas escalas (1 e 2) são sequenciais, onde cada uma consiste em uma sequência de instruções de várias transações em que as instruções que pertencem a uma única transação aparecem agrupadas.
- Suponha que estas duas transações sejam executadas de modo concorrente, onde o resultado alcançado deve ser o mesmo obtido em uma execução sequencial ($T_1 - T_2$), sendo preservada também a soma de **A** com **B**.



TRANSAÇÃO

ESCALA 3

T₁

T₂

leia(A);
A = A - 50;
escreva(A);

leia(A);
aux = A * 0.10;
A = A - aux;
escreva(A);

leia(B);
B = B + 50;
escreva(B);

leia(B);
B = B + aux;
escreva(B);

A=315
B=285

→ 600

ESCALA 4

T₁

T₂

leia(A);
A = A - 50;

leia(A);
aux = A * 0.10;
A = A - aux;
escreva(A);

leia(B);
B = B + 50;
escreva(B);

B = B + aux;
escreva(B);

A=350
B=240

→ 590

ESCALA 4 ERRADA

TRANSAÇÃO

- Assegura-se a consistência do BD, sob execução concorrente, garantindo que qualquer escala concorrente executada tenha o mesmo efeito (ou resultado) de sua execução sequencial.

Assim, o sistema de BD deve **controlar a execução concorrente de transações** para assegurar que o estado do BD **permaneça consistente**, como se todas as suas transações fossem executadas serializadas (processo de **serialização** seria garantido).



Referência de Criação e Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- ELMASRI, R. e NAVATHE, S. B., Fundamentals of Database Systems - livro
 - Capítulo 19
- SILBERSCHATZ, A., KORTH, H. F., Sistemas de Banco de Dados - livro
 - Capítulo 13

