

Transações

Exemplo

- *Você vai até o caixa eletrônico de seu banco, solicita que seja transferido um determinado valor de sua conta corrente para sua poupança.*
- *De repente, o terminal apaga!*
 - *Será que o dinheiro já saiu de minha conta?*
 - *Se saiu, será que já foi registrado na poupança???*

Transferência bancária - \$ 50

- Diversas operações mais simples para viabilizar a operação de transferência de fundos

```
read(A);
A:= A - 50;
write(A);
read(B);
B:=B+50;
write(B);
```

A - Conta corrente

B - Poupança

Exemplo

- **read(x)**: transfere o item **x** do BD para um buffer local
- **write(x)**: transfere o item **x** do buffer local de volta para o BD

Exemplo

- É desejável que ocorram **todas** essas operações mais simples, de modo a efetivar a transferência, ou **em caso de falha**, que **nenhuma** operação mais simples seja efetivada!

— Caso contrário:

- O dinheiro sai da conta corrente mas não chega na conta poupança ?!

Transações

- Conjunto de operações mais simples que devem ser todas realizadas ou todas descartadas: **Transações!**
- **Transação**: unidade de execução do programa que acessa e possivelmente atualiza vários itens de dados

Propriedades

- Para garantir a integridade dos dados é necessário que o sistema de banco de dados mantenha as propriedades **ACID** das transações

Atomicidade

Consistência

Isolamento

Durabilidade

Consistência

- Consistência:** Uma transação será preservadora de consistência se sua execução completa fizer o banco de dados passar de um estado consistente para outro.

Consistência: transferência bancária

- Consistência:** soma de $A+B$ seja inalterada pela execução da transação
 - o dinheiro não pode ser criado ou destruído pela transação!

```
T1: read(A);
    A:= A - 50;
    write(A);
    read(B);
    B:=B+50;
    write(B);
```

Consistência: transferência bancária

- Supondo que os valores iniciais de A e B, sejam respectivamente 100 e 50:
 - antes de iniciar a transação: $A+B = 150$

read(A);	100
A:= A - 50;	A:= 100 - 50
write(A);	A:= 50
read(B);	50
B:=B+50;	B:= 50 + 50
write(B);	B:= 100

- Depois de terminar a transação: $A+B = 150$

Atomicidade

- Atomicidade:** Uma transação é uma unidade atômica de processamento: ou ela será executada em sua totalidade ou não será de modo nenhum
 - todas as operações da transação são refletidas corretamente no banco de dados, ou nenhuma delas;

Atomicidade: transferência bancária

- Suponha que, durante a execução da transação T_1 , aconteça uma falha (falta de energia elétrica, falhas de hardware, etc) que impede T_1 de completar sua execução com sucesso:

read(A);	100
A:= A - 50;	A:= 100 - 50
write(A);	A:= 50
read(B);	50
B:=B+50;	B:= 50 + 50
write(B);	B:= 100 /// O novo valor é perdido!!!

Resultado: sumiram os R\$ 50,00 que foram retirados de A (valor atual de 50) mas que não foram computados em B (valor atual igual ao inicial, ou seja, manteve os 50). A soma $A+B$ não é mais preservada! **Estado inconsistente**

Atomicidade: transferência bancária

- Se a propriedade de atomicidade estiver presente, todas as ações da transação são refletidas no banco de dados ou nenhuma delas é refletida.
 - ou consegue realizar a transação como um todo ou desfaz o que estava feito até o momento da falha!
- O sistema de banco de dados deve acompanhar os valores antigos de quaisquer dados em que uma transação realiza uma escrita e, se a transação não completar com sucesso sua execução, então o sistema restaura os valores antigos para que pareça que a transação nunca sequer começou.

Durabilidade

- **Durabilidade:** As mudanças aplicadas ao banco de dados por uma transação efetivada devem permanecer no banco de dados. Essas mudanças não devem ser perdidas em razão de uma falha.

Isolamento

- **Isolamento:** Uma transação deve ser executada como se estivesse isolada das demais: a execução de uma transação não deve sofrer interferência de quaisquer outras transações concorrentes.

Isolamento: transferência bancária

- Mesmo que as propriedades de consistência e atomicidade sejam garantidas em cada transação, se várias transações forem executadas simultaneamente, suas operações podem intercalar de alguma maneira indesejável, resultando em um estado inconsistente

Isolamento: transferência bancária

- T_1 : transfere \$50 da conta A para a conta B
- T_2 : transfere 10% do saldo da conta A para a conta B.

T_1	T_2	
read(A);		A = 100 B = 50
A := A - 50;		
write(A);		
read(B);		
B := B + 50;		
write(B);		A = 50 B = 100
	read(A);	
	temp := A * 0,1;	temp = 5
	A := A - temp	
	write(A);	
	read(B);	
	B := B + temp;	
	write(B);	A = 45 B = 105

Isolamento: transferência bancária

- No exemplo anterior, uma transação é executada após o término da execução da outra: **Schedule serial**
 - Dessa forma, sempre se garante a integridade do banco
 - Se efetuarmos a soma de A+B, podemos notar que o valor é preservado.
 - Antes de iniciar: A = 100 e B = 50
 - Depois de terminar: A = 45 e B = 105

Isolamento: Schedule simultâneo

- T_1 : transfere \$50 da conta A para a conta B
- T_2 : transfere 10% do saldo da conta A para a conta B.

T_1	T_2	A = 100 B = 50
read(A);		
A := A - 50;		
write(A);		A = 50 B = 50
	read(A);	
	temp := A * 0,1;	temp = 5
	A := A - temp	
	write(A);	A = 45 B = 50
read(B);		
B := B + 50;		
write(B);		A = 45 B = 100
	read(B);	
	B := B + temp;	
	write(B);	A = 45 B = 105

Isolamento: Schedule simultâneo

- O schedule simultâneo do slide anterior é equivalente ao schedule serial apresentado anteriormente.
 - Esse schedule é dito equivalente pois reproduz exatamente o mesmo resultado que o anterior (serial).
 - Portanto, o estado do banco de dados é consistente.
- Porém, o schedule simultâneo pode levar a resultados indesejáveis, como no exemplo seguinte, onde a soma de A+B não é preservada pela execução das 2 transações

Isolamento: Schedule simultâneo

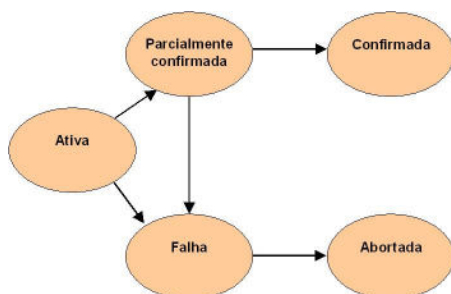
- T_1 : transfere \$50 da conta A para a conta B
- T_2 : transfere 10% do saldo da conta A para a conta B.

T_1	T_2	A = 100 B = 50
read(A);		
A := A - 50;		
	read(A);	Lê A=100
	temp := A * 0,1;	temp = 10
	A := A - temp	
	write(A);	Escreve A = 90 no BD
	read(B);	
write(A);		Escreve A = 50 no BD
read(B);		
B := B + 50;		
write(B);		Escreve B = 100 no BD
	B := B + temp;	
	write(B);	Escreve B = 60 no BD

Isolamento: transferência bancária

- No exemplo anterior, o schedule simultâneo gera inconsistências no BD
 - soma de A+B não é preservada pela execução das 2 transações
 - Se efetuarmos a soma de A+B, podemos notar que o valor é preservado.
 - Antes de iniciar: A = 100 e B = 50
 - Depois de terminar: A = 50 e B = 60
 - \$ 40 sumiram

Estado de transação



Estado de transação

- **Ativa**: é o estado inicial; a transação permanece nesse estado enquanto está sendo executada;
- **Parcialmente confirmada**: depois que a instrução final foi executada;
- **Falha**: depois da descoberta de que a execução normal não pode mais prosseguir;
- **Abortada**: depois que a transação foi revertida e o BD foi restaurado ao seu estado anterior ao início da transação;
- **Confirmada**: após o término bem sucedido.