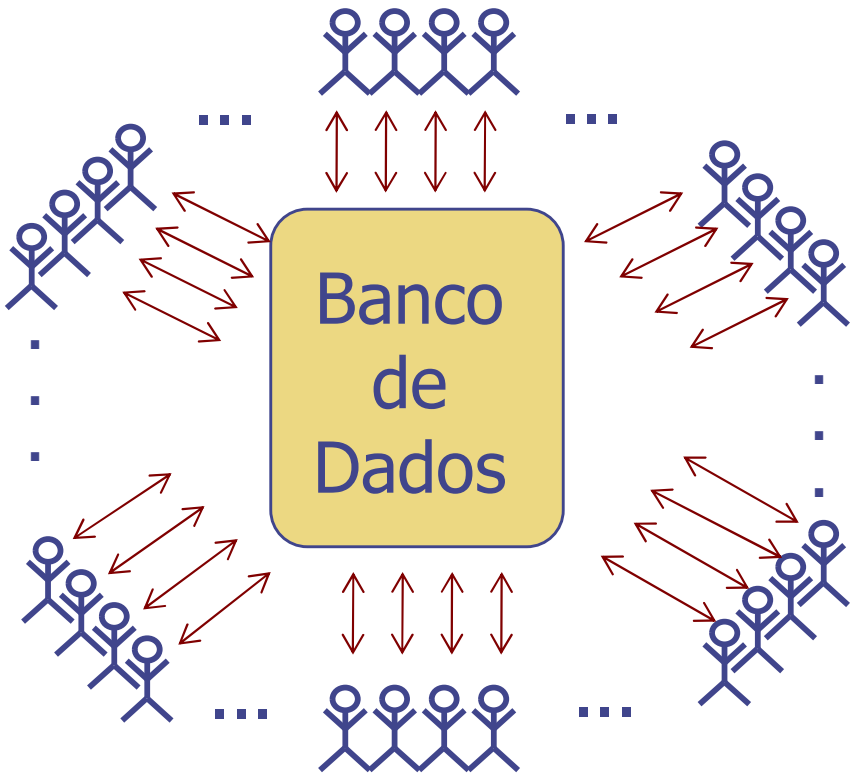
Processamento de Transações

Prof. Humberto Razente

Bloco B - sala 1B144

Introdução

- SGBDs são em geral multi-usuários
  - processam ***simultaneamente*** operações disparadas por vários usuários
  - deseja-se **alta disponibilidade** e **tempo de resposta pequeno**



- Exemplos:
- Sistemas para reserva de passagens
- Banco
- Processamento de cartões de crédito
- Sistemas de compra coletiva
- etc.

Propriedades desejáveis (ACID)

- Requisitos que sempre devem ser atendidos por uma transação para garantir a integridade dos dados
  - Atomicidade
  - Consistência
  - Isolamento
  - Durabilidade

Propriedades desejáveis

- Para garantir a integridade dos dados
  - Atomicidade: todas as operações da transação são refletidas corretamente no BD, ou nenhuma delas
  - Consistência: a execução de uma transação isolada preserva a consistência do BD
  - Isolamento: a transação não deve sofrer interferência de nenhuma outra transação concorrente
  - Durabilidade: as alterações realizadas por uma transação que completou com sucesso são persistidas mesmo que ocorram falhas no sistema

Atomicidade

- Princípio do *"Tudo ou Nada"*
  - ou todas as operações da transação são efetivadas com sucesso no BD ou nenhuma delas se efetiva
    - preservar a integridade do BD
  - Responsabilidade do subsistema de recuperação contra falhas (subsistema de *recovery*) do SGBD
    - desfazer as ações de transações parcialmente executadas

Consistência

- Uma transação sempre conduz o BD de um estado consistente para outro estado também consistente
  - durante a execução da transação, a base de dados pode passar por um estado inconsistente
- Responsabilidade conjunta do
  - DBA/programador
    - definir todas as restrições de integridade (RI) para garantir estados e transições de estados válidos para os dados
  - subsistema de *recovery*
    - desfazer as ações das transações que violaram a integridade

Isolamento

- A execução de uma transação T_x deve funcionar como se T_x executasse de forma isolada
  - T_x não deve sofrer interferências de outras transações executando concorrentemente
  - Resultados intermediários das transações devem ser escondidos de outras transações executadas concorrentemente
  - Responsabilidade do subsistema de controle de concorrência do SGBD

Isolamento

T_1	T_2
read(A)	
A = A – 50	
write(A)	
	read(A)
	A = A+A*0.1
	write(A)
read(B)	
B = B + 50	
write(B)	
	read(B)
	B = B - 100
	write(B)

escalonamento válido

T_1	T_2
read(A)	
A = A – 50	
	read(A)
	A = A+A*0.1
	write(A)
	read(B)
write(A)	
read(B)	
B = B + 50	
write(B)	
	B = B - 100
	write(B)

T_1 interfere em T_2

T_2 interfere em T_1

escalonamento inválido

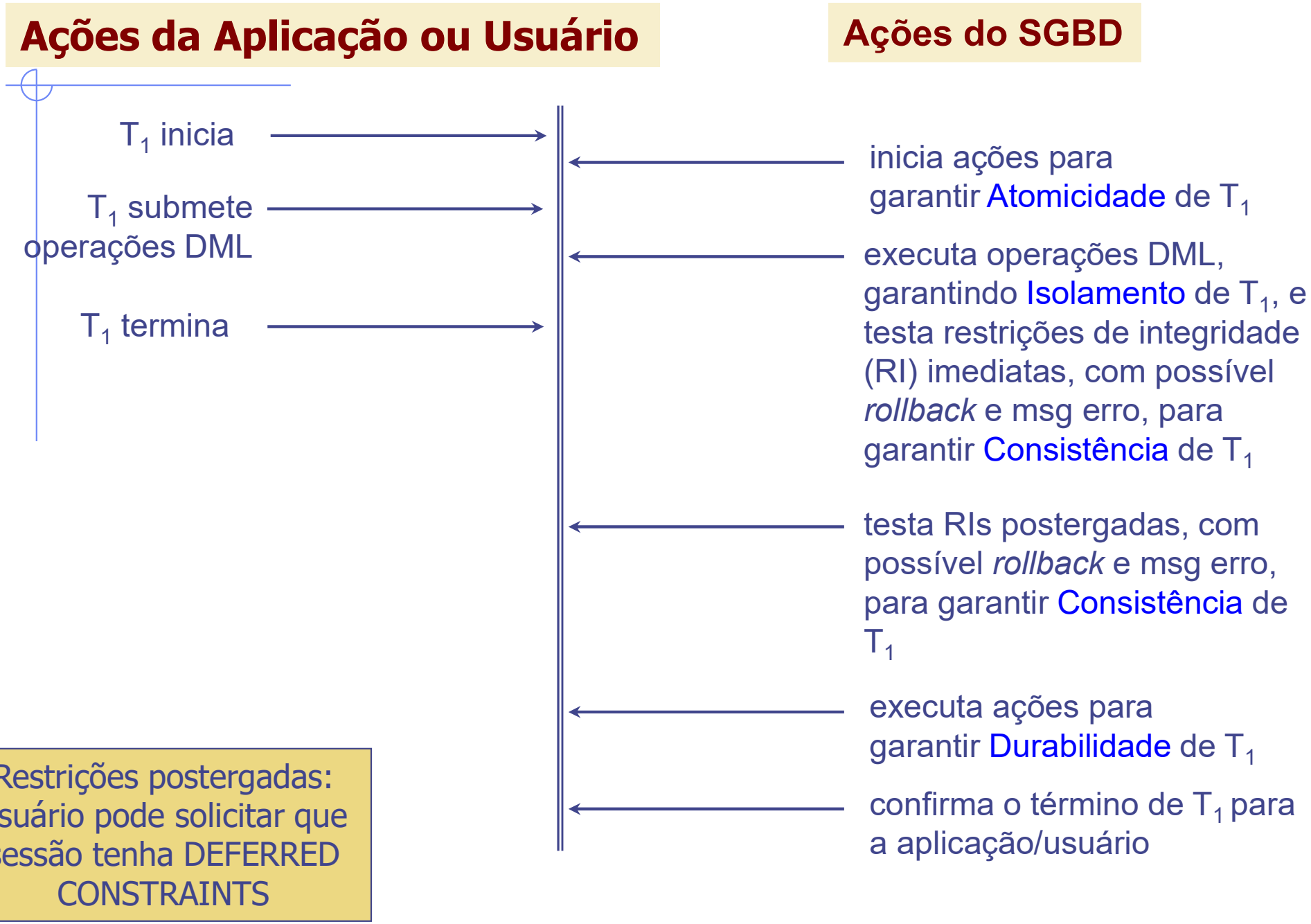
Durabilidade

- Deve-se garantir que as modificações realizadas por uma transação que concluiu com sucesso persistam no BD
  - nenhuma falha posterior ocorrida no BD deve perder essas modificações
  - Responsabilidade do subsistema de *recovery*
    - refazer transações que executaram com sucesso em caso de falha no BD

Exemplo

- Transação para transferir R$ 50 da conta ***A*** para conta ***B***
  1. Read (A)
  2. A = A – 50
  3. Write (A)
  4. Read (B)
  5. B = B + 50
  6. Write (B)
  - Requisito de ***Consistência***
    - A soma de *A* e *B* não se altera pela execução da transação (o dinheiro apenas mudou de conta)
  - Requisito de ***Atomicidade***
    - Se a transação falhar após o passo 3 e antes do passo 6, o sistema deve assegurar que as suas atualizações não sejam refletidas no banco de dados, senão resultará em uma inconsistência
  - Requisito de ***Durabilidade***
    - Uma vez que o usuário tenha sido notificado que a transação foi completada (ou seja, a transferência dos R$50 ocorreu), as atualizações no banco de dados feitas pela transação devem persistir, mesmo na ocorrência de falhas
  - Requisito de ***Isolamento***
    - Se, entre os passos 3 e 6, outra transação tiver permissão para acessar o banco de dados parcialmente atualizado, ela verá um banco de dados inconsistente (a soma A + B valerá menos do que deveria)

Gerência Básica de Transações



Restrições postergadas: usuário pode solicitar que sessão tenha DEFERRED CONSTRAINTS

Definição de transação na SQL

- A linguagem de manipulação de dados precisa incluir uma construção para especificar o conjunto de ações que compõem uma transação
- Na SQL, uma transação começa implicitamente
- Uma transação na SQL termina por:
  - Commit - confirma a transação atual e inicia uma nova transação
  - Rollback - faz com que a transação atual seja abortada
- Níveis de consistência especificados pela SQL-92:
  - Serializable → padrão no SQL-92
  - Repeatable read
  - Read committed
    - padrão em algumas implementações de SGBD
  - Read uncommitted

Níveis de consistência na SQL-92

Tabela 21.1 – Violações possíveis com base nos níveis de isolamento definidos na SQL

Nível	Leitura suja	Leitura não repetitiva	Fantasma
Read uncommitted	Sim	Sim	Sim
Read committed	Não	Sim	Sim
Repeatable read	Não	Não	Sim
Serializable	Não	Não	Não

- **Leitura suja**: uma transação T_1 pode ler a atualização de uma transação T_2, que ainda não foi confirmada. Se T_2 falhar e for abortada, então T_1 teria lido um valor que não existe e é incorreto
- **Leitura não repetitiva**: uma transação T_1 pode ler determinado valor de uma tabela. Se outra transação T_2 mais tarde atualizar esse valor e T_1 ler o valor novamente, T_1 verá um valor diferente
- **Fantasmas**: uma transação T_1 pode ler um conjunto de linhas de uma tabela, com base em uma cláusula WHERE. Se T_2 inserir uma nova linha que também satisfaça a cláusula WHERE usada em T_1. Se T_1 for repetida, T_1 verá um fantasma, uma linha que anteriormente não existia.

Leitura complementar para casa

- Capítulo 21 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados, 6ª edição