

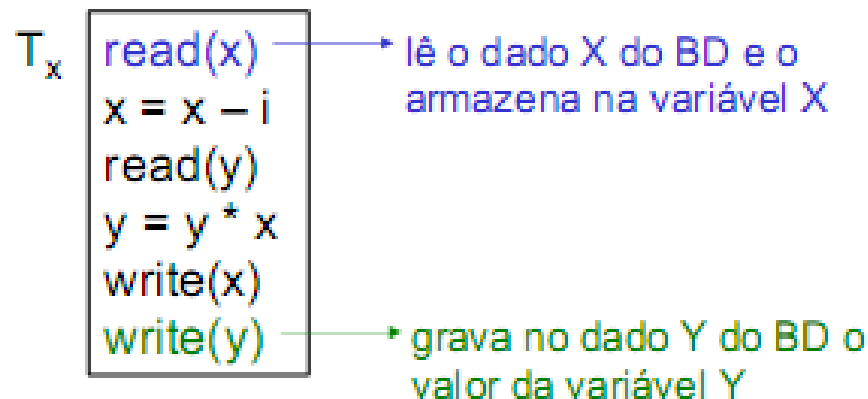
INSTITUTO FEDERAL
PIAUÍ
Campus Parnaíba

Transações

Prof. Msc Denival A. dos Santos

Transação

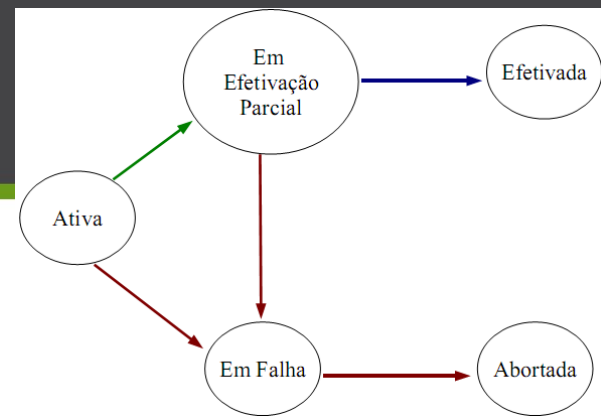
- É a unidade básica de processamento em um SGBD. Do ponto de vista do SGBD, uma transação é uma sequência de operações que são tratadas como um bloco único e indivisível (atômico) no que se refere à sua recuperação.
- Composta de uma ou mais operações delimitadas pelas declarações:
 - Início: **Begin transaction** e Termina: **Commit ou rollback transaction**
- Transação está envolvida com os comandos: **Update, Delete e Insert;**
- De forma abstrata e simplificada, uma transação pode ser encarada como um conjunto de operações de leitura e escrita de dados;



Término da transação

- É indicado pelos comandos:
 - **Commit:** Termina com sucesso uma transação
 - **Rollback:** Termina uma transação indicando que houve um erro.
- Se uma transação terminar com commit, todas as modificações realizadas dentro dela, serão efetivadas;
- Se uma transação terminar com rollback, todas as modificações realizadas dentro dela, serão desfeitas;
 - O banco de dados ficará no mesmo estado que estava antes da transação ser iniciada;

Estados de uma transação



- **Ativa** - Estado inicial da transação. Ela permanece neste estado enquanto estiver em execução;
- **Em efetivação parcial** - Após a execução da última declaração;
- **Em falha** - Após a descoberta de que a execução normal já não pode ser realizada;
- **Abortada** - Depois que a transação foi desfeita e o banco de dados foi restabelecida ao estado anterior do início da execução da transação;
- **Efetivada** - Após a conclusão com sucesso.

Propriedade das transações

- As transações devem ser executadas pelos SGBDs de maneira a evitar que problemas ocorram;
- As ações que compõem uma transação devem possuir um conjunto de propriedades que é normalmente referido como propriedades ACID, que são:
 - **A**tomicidade
 - **C**onsistência
 - **I**solamento
 - **D**urabilidade

Atomicidade

- Esta propriedade garante que todas as operações na transação serão executadas ou nenhuma será;
- Isto evita que falhas ocorridas possam deixar o banco de dados inconsistente;
- O exemplo clássico para a necessidade de uma "transação atômica" é aquele da transferência entre duas contas bancárias. No momento de uma transferência de valores de uma conta "A" para uma conta "B", que envolve pelo menos uma operação de ajuste no saldo para cada conta, se o computador responsável pela operação é desligado por falta de energia, é esperado que o saldo de ambas as contas não tenha se alterado;
- Tipicamente, esse conceito é implementado por algum mecanismo que indica que a transação foi iniciada ou terminou, ou pela manutenção de uma cópia dos dados feita antes do início da transação;
- Os SGBDs utilizam sistemas de log de dados que gravam os dados ainda não confirmados. No caso da confirmação da transação (commit), esses dados são consolidados na base de dados, por outro lado, na ocorrência de algum problema (rollback) esses dados são descartados.

Consistência

- Busca dois fatores:
 - A consistência dos banco de dados
 - A consistência da própria transação
- Uma transação deve ser um programa correto e suas ações não devem resultar em violações das restrições de integridade definidas para o banco de dados;
- Uma transação sempre conduz o BD de um estado consistente para outro estado também consistente;
- Responsabilidade conjunta do:
 - DBA
 - Definir todas as regras de integridade para garantir estados e transições de estados válidos para os dados:
 - Exemplo : $\text{salario} > 0$, $\text{Salario_novo} > \text{salario_antigo}$;
 - Subsistema de recovery (recuperação)
 - Desfazer as ações da transação que violou a integridade.

Isolamento

- Significa que, mesmo no caso de transações executadas concorrentemente, o resultado final é igual ao obtido com execução de cada uma delas;
- A observância desta propriedade das transações pelos SGBDs impede a ocorrência dos problemas de acesso a dados;
- No contexto de um conjunto de transações concorrentes, a execução de uma transação Tx deve funcionar como se Tx executasse de forma isolada;
 - Tx não deve sofrer interferência de outras transações executando concorrentemente
- Responsabilidade do subsistema de controle de concorrência (scheduler) do SGBD - garantir escalonamento sem interferências.

Isolamento - Exemplo

| 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 - A 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 - A write(B) ← |

T₁ interfere em T₂

T₂ interfere em T₁

escalonamento inválido

Durabilidade ou persistência

- Significa que os resultados de uma transação, caso ela seja concluída com sucesso, devem ser persistentes. Mesmo se depois houver uma falha no sistema;
- 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.

Gerência básica de transações

Ações da Aplicação ou Usuário

T_1 inicia →
 T_1 submete operações DML →
 T_1 termina →

Ações do SGBD

← inicia ações para garantir **Atomicidade** de T_1
← executa operações DML, garantindo **Isolamento** de T_1 , e testa RIs imediatas, com possível *rollback* e msg erro, para garantir **Consistência** de T_1
← testa RIs postergadas, com possível *rollback* e msg erro, para garantir **Consistência** de T_1
← executa ações para garantir **Durabilidade** de T_1
← confirma o término de T_1 para a aplicação/usuário

Transações no *MySQL*

- Início das transações
Start transaction
- Término da transação
Commit (com sucesso)
Rollback (fracasso)