

T07: Transações e Concorrência

Transações

Requisitos (ACID)

Estados de Transação

Execuções Concorrentes

Esquemas de Controlo de Concorrência

Escalonamento

Serialização

Testando Serialização

Protocolos Baseados em Trincos

Matriz de Compatibilidade de Trincos

Transações



Uma **transação** é uma unidade de execução de um programa que acede e (possivelmente) atualiza vários dados.

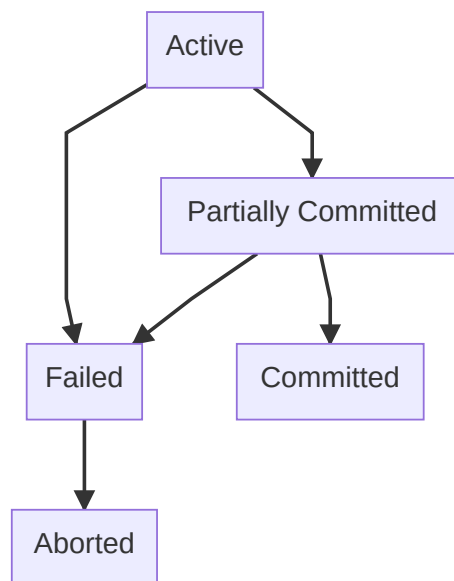
Requisitos (ACID)

- **Atomicidade:**
 - Uma transação tem de ser tratada como uma unidade atômica;
 - Todas as suas operações são executadas ou nenhuma é executada;
 - Não deve haver nenhum estado numa BD em que uma transação seja deixada parcialmente concluída.
- **Consistência:**
 - A BD tem de estar num estado consistente após qualquer transação;
 - Nenhuma transação deve ter qualquer efeito adverso sobre os dados armazenados.
- **Durabilidade:**
 - A BD deve ser durável o suficiente para conter todas as suas atualizações mais recentes, mesmo se o sistema falhar;
 - Se uma transação atualiza parte dos dados e confirma, a BD manterá os dados modificados.

- **Isolamento:**

- Num SGBD onde mais de uma transação está a ser executada em paralelo, esta propriedade afirma que todas as transações serão realizadas e executadas como se fossem a única transação no sistema;
- Nenhuma transação afetará a existência de qualquer outra transação.

Estados de Transação



Execuções Concorrentes

- Várias transações podem ser executadas simultaneamente no sistema:
 - aumenta a utilização do disco e do CPU, levando a um melhor *throughput* de transações;
 - redução do tempo médio de resposta para transações;

Esquemas de Controlo de Concorrência

- Mecanismos que alcançam o isolamento:
 - Controlam a interação entre as transações concorrentes, a fim de impedi-los de destruir a consistência da BD;

Escalonamento

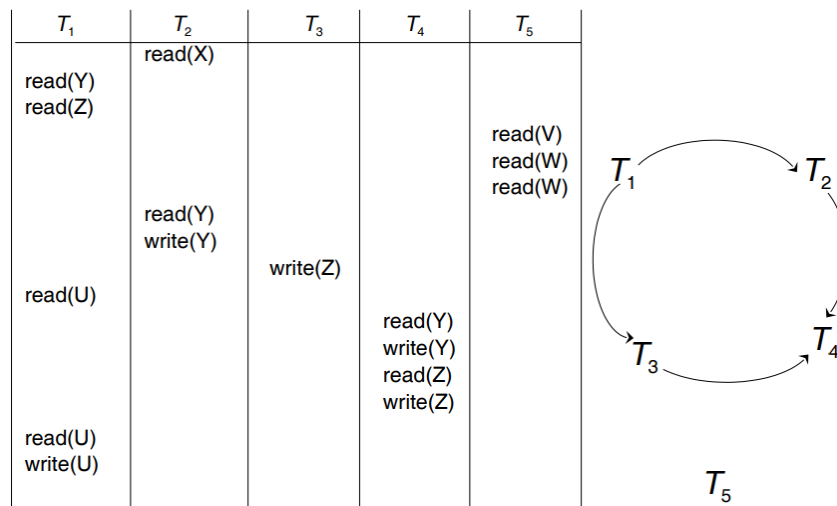
- Sequência de instruções que especificam a ordem cronológica na qual instruções de transações concorrentes são executadas:
 - Um escalonamento para um conjunto de transações consiste em todas as instruções dessas transações;
- Uma transação que conclui sua execução com sucesso terá a instrução **COMMIT** como a última declaração;
- Uma transação que falha em concluir a sua execução com sucesso terá a instrução **ABORT** como a última declaração;

Serialização

- Uma escalonamento (possivelmente concorrente) é serializável se for equivalente a um escalonamento em série;
- Diferentes formas de concorrência equivalente dão origem às noções de:
 - **Conflict Serializability**: um escalonamento é serializável por conflito se puder ser transformado num escalonamento serial por meio da troca de operações não conflitantes;
 - Duas operações são consideradas conflitantes se todas as condições forem satisfeitas:
 - Pertencem a diferentes transações;
 - Operam sobre os mesmos dados;
 - Pelo menos uma das operações é **WRITE**.

Testando Serialização

- **Grafo de Precedência**: um grafo direcionado onde os vertices são transações;
- É feito um arco de T_i a T_j se as duas transações forem conflitantes, e T_i acedeu ao dado no qual foi originado o conflito.



Protocolos Baseados em Trincos

- Um trinco é um mecanismo utilizado para controlar acesso concorrente a dados;
- Dados podem ser "fechados" em dois modos:
 - **EXCLUSIVE** (X): dado pode ser lido e escrito;
 - **SHARED** (S): dado pode ser apenas lido.

Matriz de Compatibilidade de Trincos

MODO	S	X
S	TRUE	FALSE
X	FALSE	FALSE