



SCC-240/540/640 Bases de Dados

Prof. Robson L. F. Cordeiro

Aula 17 – Transações

Material original editado: Profa. Elaine Parros Machado de Sousa

Transações

- **Transação**: Unidade lógica de trabalho
 - abrange um conjunto de operações de manipulação de dados que executam uma única tarefa

Conecta ao Banco de Dados

Começa transação

Operações de consulta/atualização

...

Finaliza transação

Começa transação

Operações de consulta/atualização

...

Aborta transação

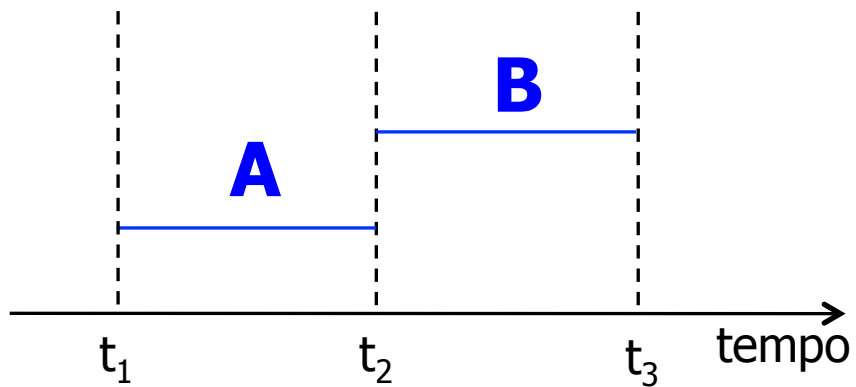
Desconecta

Transações

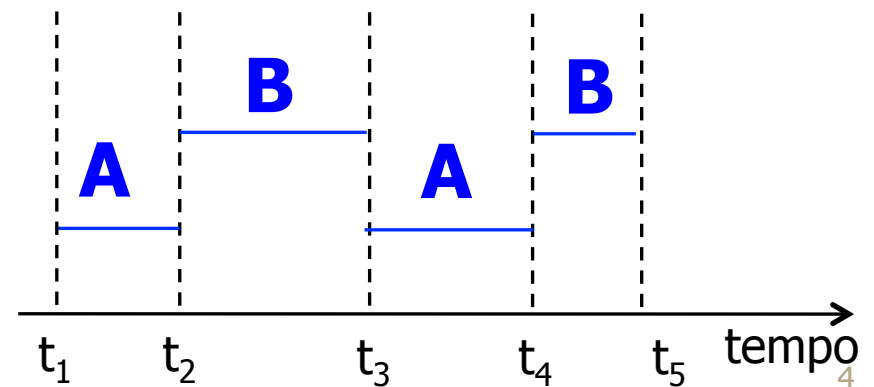
- SGBDs multi-usuários
 - vários usuários/aplicações usando o SGBD concorrentemente
 - mesmo usuário/aplicação pode conter várias transações
 - transações concorrentes
 - transações podem acessar/atualizar os mesmos dados
 - **granularidade** dos dados (**item de dado**): campo, tupla, tabela, base de dados

Transações

Execução Serial



Execução Intercalada



Transações

- Quando uma transação começa?
 - explicitamente com a operação de **begin_transaction**
 - implicitamente quando uma sessão é iniciada no SGBD
 - implicitamente após a execução de um comando **DDL**, **commit** ou **rollback**

Transações

- Quando uma transação termina?
 - explicitamente com **commit** ou **rollback**
 - implicitamente quando um processo de usuário é finalizado
 - com sucesso – ex: disconnect (**commit**)
 - sem sucesso – ex: falha de sistema (**rollback**)
 - usuário (aplicação) executa comando DDL

Transações

- O que acontece quando duas transações executam simultaneamente manipulando o mesmo item de dado?
 - O que acontece se a energia acabar no meio de uma transação, ou se houver um problema com o disco?
- ⇒ O banco de dados pode ser levado a um estado inconsistente...



Transações – **Propriedades ACID**

Atomicidade

Consistência

Isolamento

Durabilidade

Transações – Propriedades ACID

- **Atomidade**: todas as operações de uma transação devem ser efetivadas. Ou, na ocorrência de uma falha, nada deve ser efetivado
 - “tudo ou nada”
- **Consistência**: transações preservam a consistência da base
 - Estado inicial consistente \Rightarrow Estado final consistente

Transações – Propriedades ACID

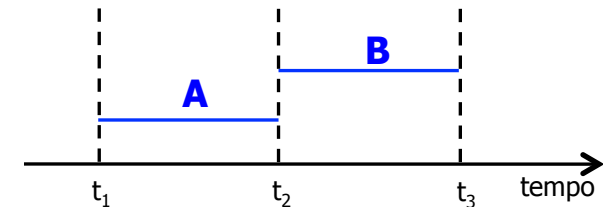
- **Isolamento**: uma transação A não vê o efeito de uma transação B até que B termine
- **Durabilidade**: uma vez terminada a transação, as alterações realizadas permanecem no banco até que outras alterações sejam explicitamente realizadas

Garantindo as Propriedades ACID

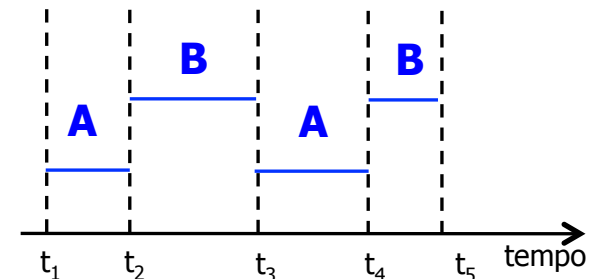
- *Logging* e Recuperação de falhas
 - Garantem a **Atomicidade** e a **Durabilidade**
 - Módulos de *Log*
- Controle de Concorrência
 - Garante a **Consistência** e o **Isolamento**, (dada a atomicidade das transações)
 - Módulos de *Locks*

Controle de Concorrência

- **Execução Serial:** diversas transações executadas em sequência
 - deixa a base de dados em estado correto e consistente
 - maior isolamento
 - menor concorrência



- **Execução Intercalada:** comandos de diversas transações são intercalados
 - pode levar a inconsistências
 - menor isolamento
 - maior concorrência



Execução Serial X Intercalada

- **Execução serial**
 - estado final da base de dados após **execução serial pode variar dependendo da ordem em que as transações são executadas**
 - mas **todos são estados corretos e consistentes**

Execução Serial X Intercalada

- **Execução Intercalada**
 - Estado final da base de dados após **execução intercalada** é consistente se for igual ao resultado obtido por uma (qualquer) execução serial
 - esta execução é dita **SERIALIZÁVEL**

Controle de Concorrência

Por que CONTROLE DE CONCORRÊNCIA???

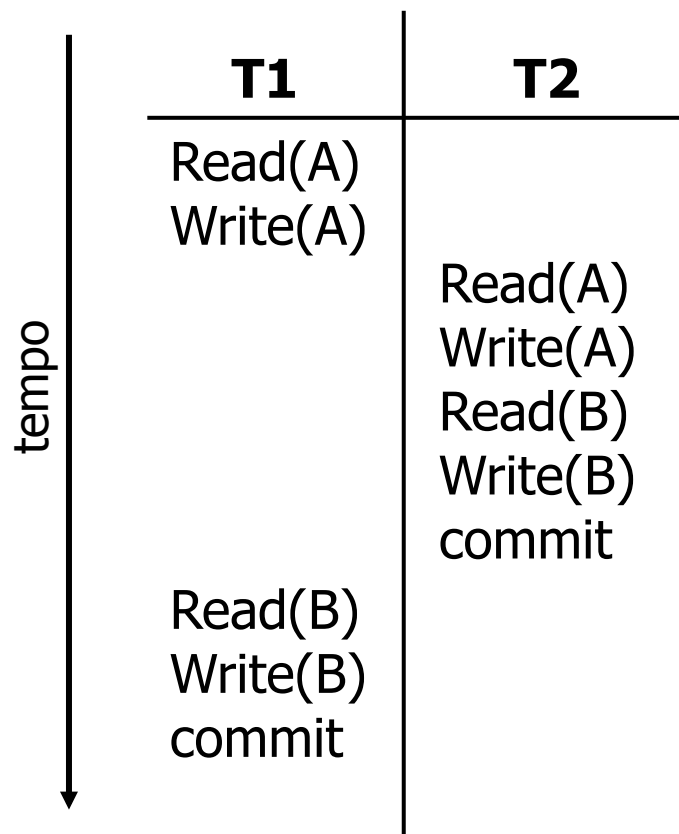
- Dependendo da ordem em que os comandos de duas ou mais transações são intercalados:
 - o resultado final pode ser inconsistente \Rightarrow ou seja, não existe uma sequência de transações executadas em série que leve a esse resultado
- Ocorrência de anomalias.
 - leitura inválida
 - leitura não repetível
 - leitura fantasma
 -

Anomalias

- **Leitura inválida (*Dirty Read*):**
 - transação T_2 lê um dado modificado por uma transação T_1 que ainda não terminou.

Anomalias

- Ex: Leitura inválida (*Dirty Read*):



Exemplo 1:

- **Transação T_1** : transfere R\$100,00 da conta A para a conta B.
- **Transação T_2** : incrementa A e B em 1% (juros).

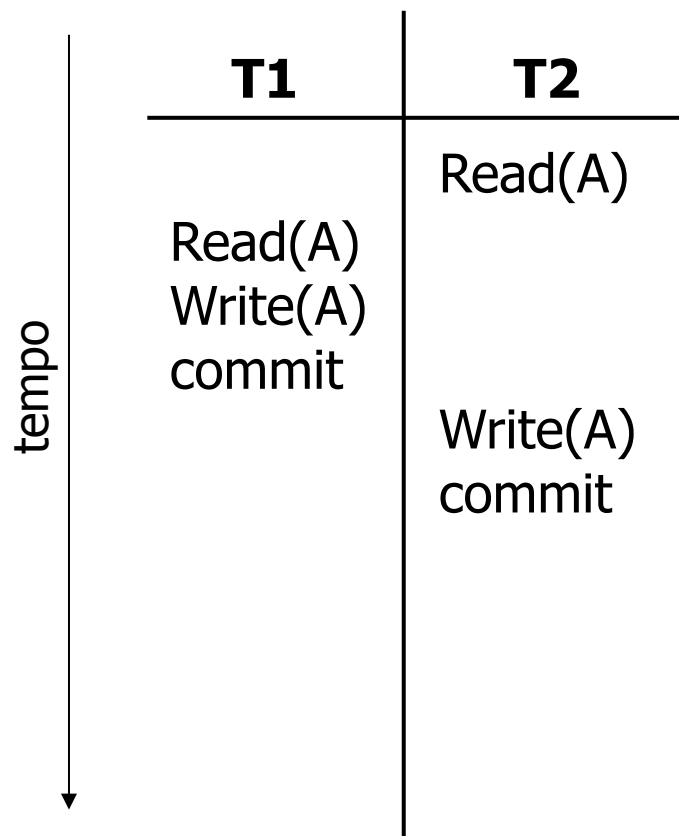
Execução não serializável ⇒
estado final inconsistente!

Anomalias

- **Leitura não repetível (*Nonrepeatable Read*):**
 - transação T_2 lê um dado
 - esse dado é modificado por uma transação T_1 que começou depois de T_2
 - T_1 é efetivada
 - se T_2 tentar reler o mesmo dado, terá dois valores diferentes (*nonrepeatable read*)

Anomalias

- Ex: Leitura não repetível (*Nonrepeatable Read*):



Exemplo:

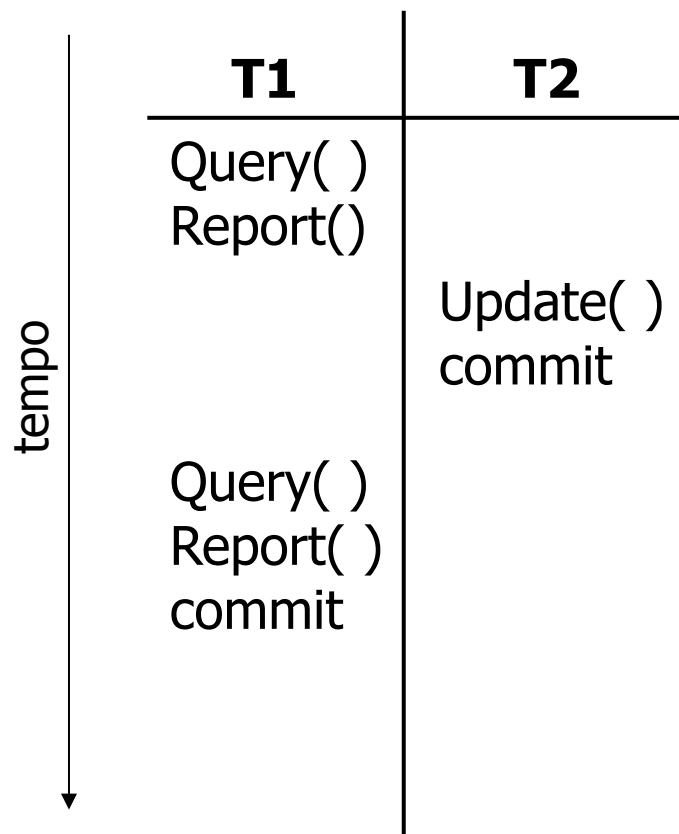
- **Transação T₂**: lê reservas de um voo e verifica que há apenas um lugar disponível.
- **Transação T₁**: lê a mesma coisa
- **T₁** reserva o último lugar e é efetivada.
- **T₂** tenta reservar o lugar e gera uma inconsistência

Anomalias

- **Leitura fantasma (*Phantom Read*):**
 - transação T_1 lê um conjunto de tuplas que atendam a uma condição de consulta
 - transação T_2 insere/remove/atualiza uma tupla que atenderia a essa condição e é efetivada
 - se T_1 refizer a mesma consulta, obterá um conjunto diferente de tuplas (*phantom read*)

Anomalias

- Ex: Leitura fantasma (*Phantom Read*):



Exemplo:

- **Transação T₁**: faz uma consulta que retorna a média geral dos alunos que têm média ponderada acima de 5.0, e gera um relatório
 - **Transação T₂**: atualiza as notas de alguns alunos e é efetivada
 - **T₁** refaz a consulta para gerar relatório com nro de alunos por faixa de média
- ⇒ **relatórios inconsistentes.**

Níveis de Isolamento em SQL99

Nível de isolamento	Anomalias que PODEM ocorrer		
	Leitura inválida	Leitura não repetível	Leitura fantasma
Read uncommitted	Sim	Sim	Sim
Read committed	Não	Sim	Sim
Repeatable read	Não	Não	Sim
Serializable	Não	Não	Não

Exercício

- Considere as transações T1 e T2, executadas sobre os itens de dado X e Y

<p>T1 Read(X) Read(Y) Write(X) commit</p>
--

<p>T2 Read(X) Read(Y) Write(X) Write(y) commit</p>

- 1) Dê um exemplo de execução intercalada que resulte em uma anomalia de leitura inválida e explique o porquê.
- 2) Dê um exemplo de execução intercalada que resulte em uma anomalia de leitura não repetível e explique o porquê.