

Transações

Transações

- **Transação:** Unidade lógica de trabalho
 - É 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

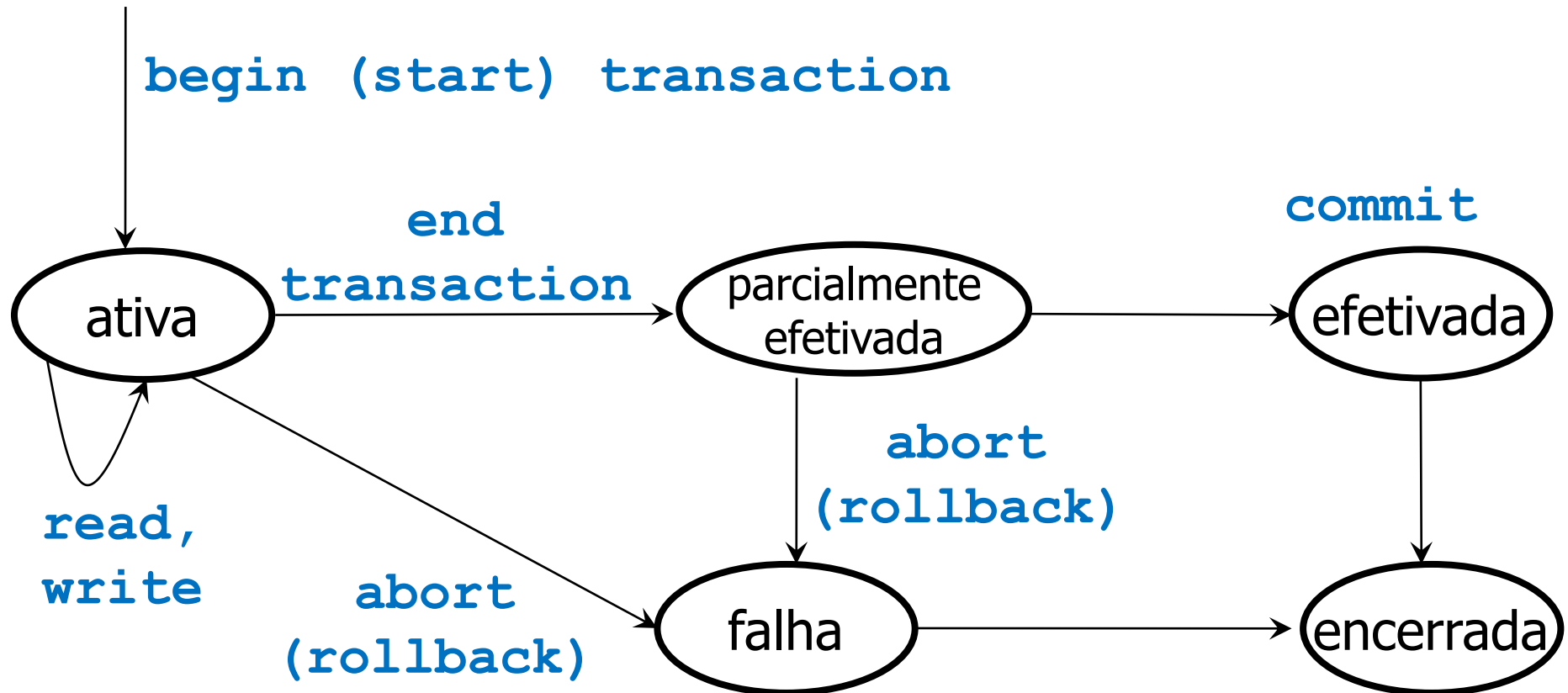
...

Finaliza transação

Desconecta

Transações

- Transição de Estados de Execução



Transações

- O que acontece se a energia acabar no meio de uma transação, ou se houver um problema com o disco?
- O que acontece quando duas transações executam simultaneamente manipulando o mesmo dado?

⇒ O banco de dados pode ser levado a um estado inconsistente...

Falhas

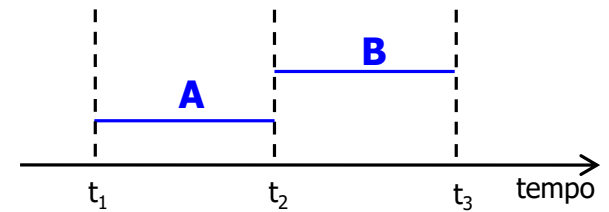
	Tipo de falha	Abrangência	Como evitar/recuperar
Local	Local	Apenas a transação corrente	Registro de log/control de concorrência
Global	De sistema (soft crash)	Todas as transações em andamento	Registro de log
Global	De meio físico (hard crash)	Todas as transações em andamento	Backup

Transações – Propriedades ACID

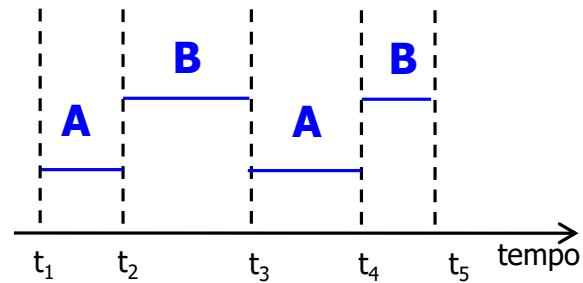
- **Atomicidade**: todas as operações de uma transação devem ser efetivadas; ou, na ocorrência de uma falha, nada deve ser efetivado
 - “**tudo ou nada**” – não se admite parte de uma operação
- **Consistência**: transações preservam a consistência da base
 - Estado inicial consistente \Rightarrow Estado final consistente
- **Isolamento**: a maneira como várias transações em paralelo interagem (o que pode ser lido e o que pode ser escrito por cada uma) deve ser bem definido
- **Durabilidade**: uma vez consolidada (committed) a transação, suas alterações permanecem no banco até que outras transações aconteçam

Controle de Concorrência

- Execução Serial (sequencial):









- Execução Intercalada:



Controle de Concorrência

- **Execução Serial (sequencial):** diversas transações executadas em sequência
 - deixa a base de dados em estado correto e consistente
- **Execução Intercalada:** comandos de diversas transações são intercalados
 - pode levar a inconsistências

	Isolamento	Concorrência	Chance de inconsistências
Serial			
Intercalada			

Execução Serial X Intercalada

- **Execução serial**

- estado inicial correto e consistente \Rightarrow estado final correto e consistente

Execução Serial X Intercalada

- **Execução Intercalada**

- Toda execução serial é consistente
- Mas uma execução intercalada só é consistente se for igual ao resultado de uma execução em sequência (em ordem conhecida)
 - esta execução é dita **serializável**

Problemas de Execução Intercalada

- Ocorrência de anomalias
 1. leitura inválida
 2. leitura não repetível
 3. leitura fantasma

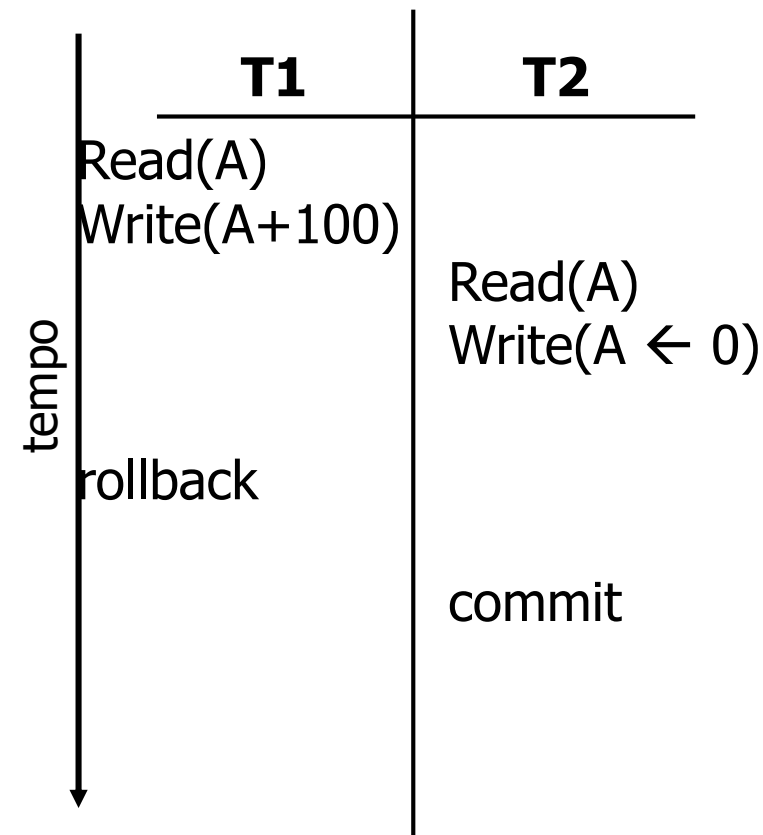
Problemas de Execução Intercalada

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

- transação T' lê um dado modificado por uma transação T que ainda não terminou;
- permite que outras transações possam ver os dados que ainda não foram consolidados (committed), isto é, mudanças que podem ser descartadas em seguida, por causa de uma instrução ROLLBACK por exemplo.

Problemas de Execução Intercalada

•Ex: Leitura inválida (*Dirty Read*):



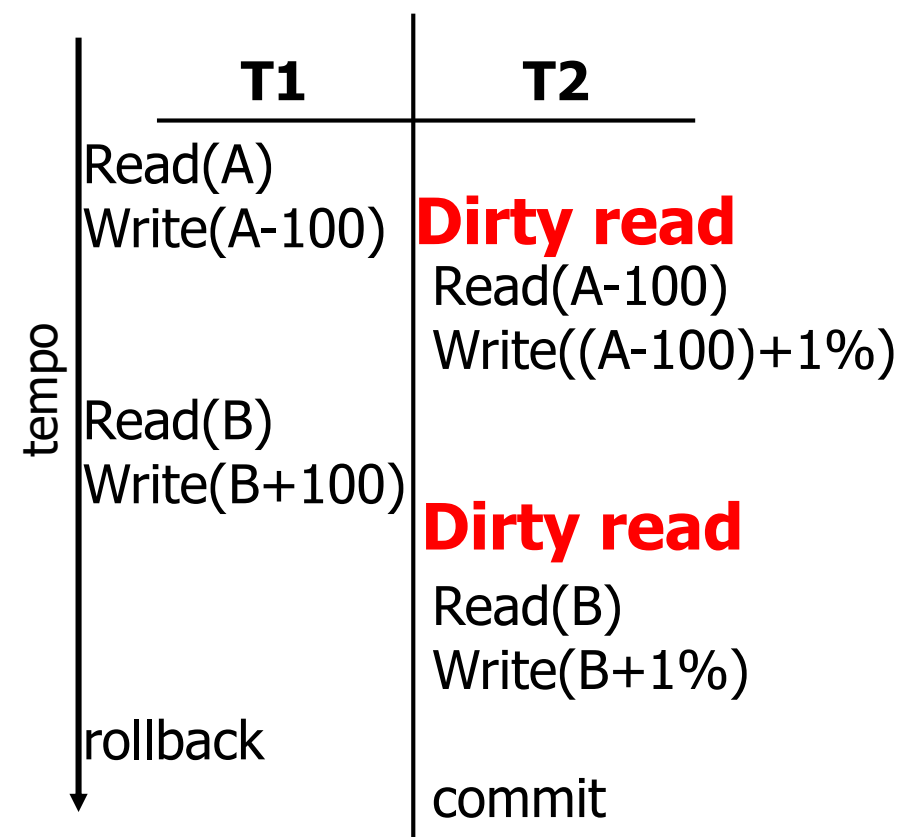
Exemplo 1:

- **Transação T_1** : deposita R\$100,00 na conta A.
- **Transação T_2** : saca tudo de A.
- T_1 é cancelada

Resultado: foi possível sacar R\$ 100,00 a mais.

Problemas de Execução Intercalada

•Ex: Leitura inválida (*Dirty Read*):



Exemplo 2:

Supondo que inicialmente $A = 500$ e $B = 600$
• resultado esperado: T1 (rollback) seguido de T2 é $A = 505$ ($A+1\%$) e $B = 606$ ($B+1\%$)

• no entanto, com T1 e T2 em paralelo, o que se tem é: $A = 404$ ($A - 100 + 1\%$) e $B = 707$ ($B+100 + 1\%$)

Resultado: a transferência de R\$ 100 de T1 foi sobrescrita e T2 considerou valores errados como base se cálculo de 1%. Ambos, A e B estão errados.

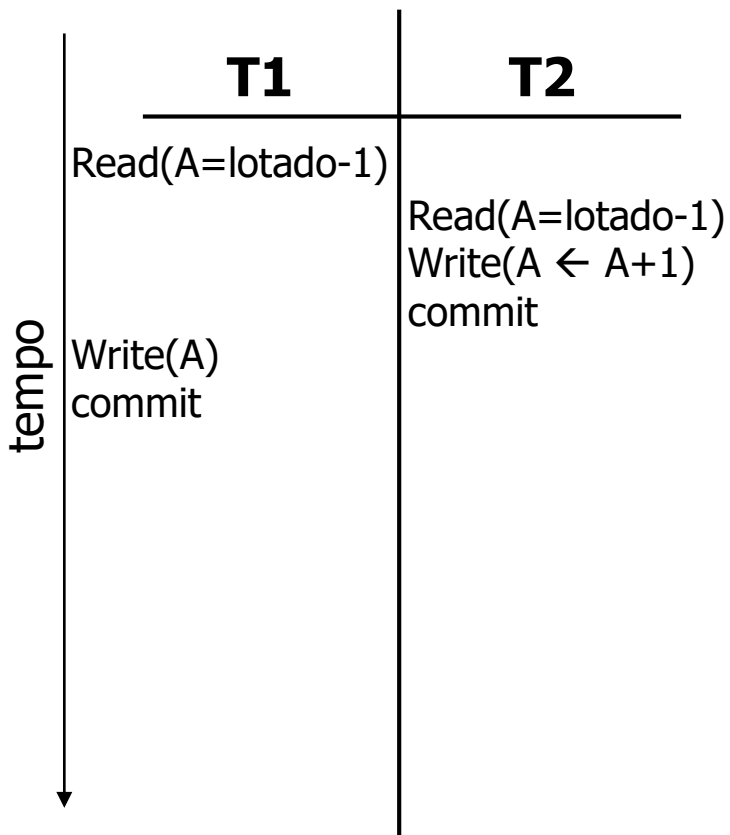
Problemas de Execução Intercalada

• **2) Leitura não repetível (*Nonrepeatable Read*):**

- transação T lê um dado
- esse dado é modificado por uma transação T' que começou depois de T
- T é efetivada
- se T' tentar reler o mesmo dado, obterá valores diferentes (*nonrepeatable read*)

Problemas de Execução Intercalada

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

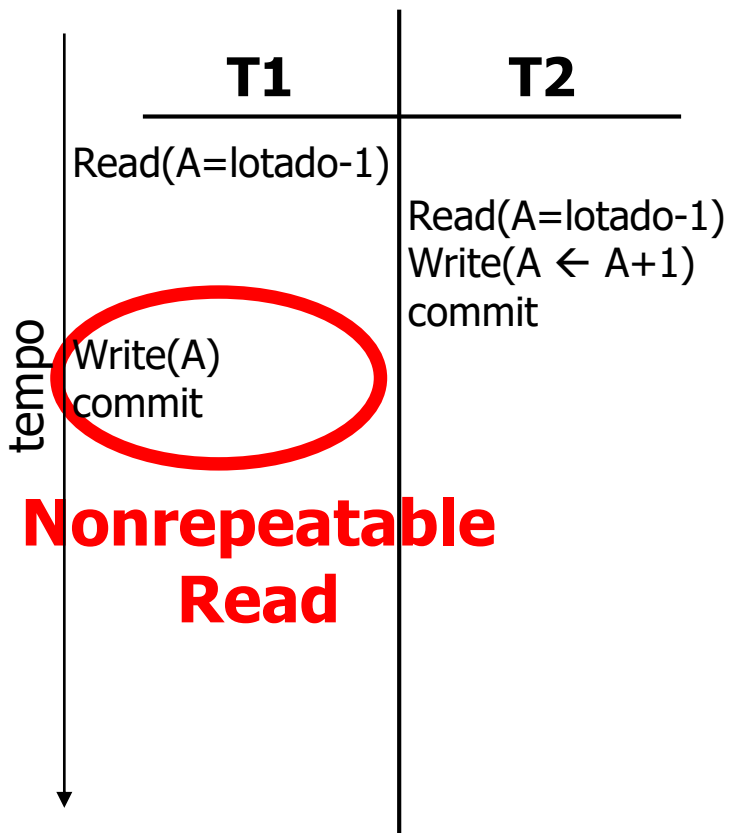


Exemplo:

- **Transação T_1** : lê reservas de um voo e verifica que há apenas um lugar disponível.
- **Transação T_2** : lê a mesma coisa
- **T_2** reserva o último lugar e é efetivada.
- **T_1** tenta reservar o lugar e ocorre um erro.

Problemas de Execução Intercalada

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

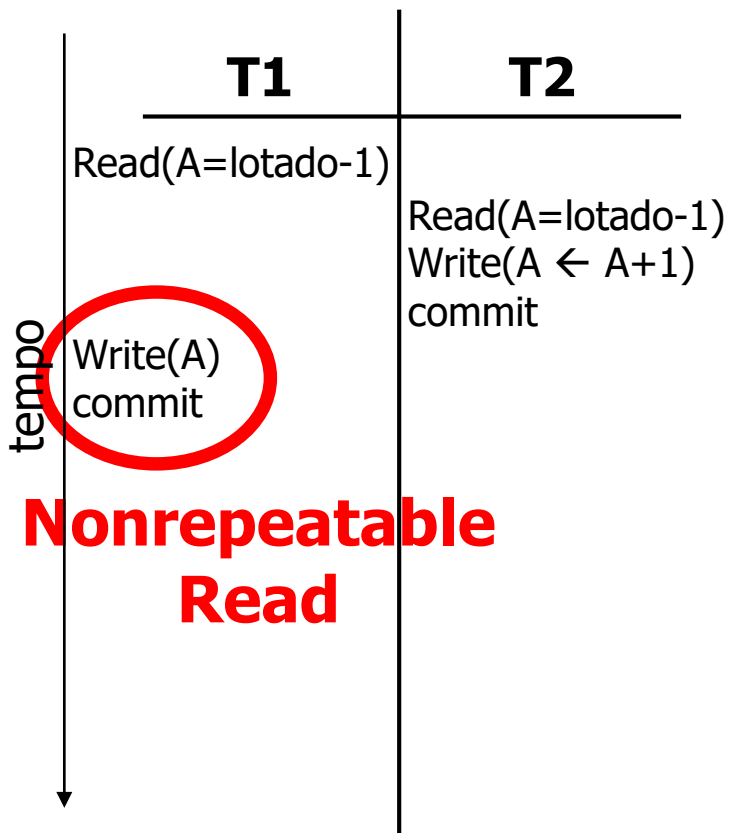


Exemplo:

- **Transação T_1** : lê reservas de um voo e verifica que há apenas um lugar disponível.
- **Transação T_2** : lê a mesma coisa
- **T_2** reserva o último lugar e é efetivada.
- **T_1** tenta reservar o lugar e ocorre um erro.

Problemas de Execução Intercalada

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



Exemplo:

- **Transação T_1** : lê reservas de um voo e verifica que há apenas um lugar disponível.
- **Transação T_2** : lê a mesma coisa
- **T_2** reserva o último lugar e é efetivada.
- **T_1** tenta reservar o lugar e ocorre um erro.

Resultado: um usuário foi informado de que ainda havia lugares. Após preencher o cadastro, clicou confirmar e recebeu um erro: voo lotado.

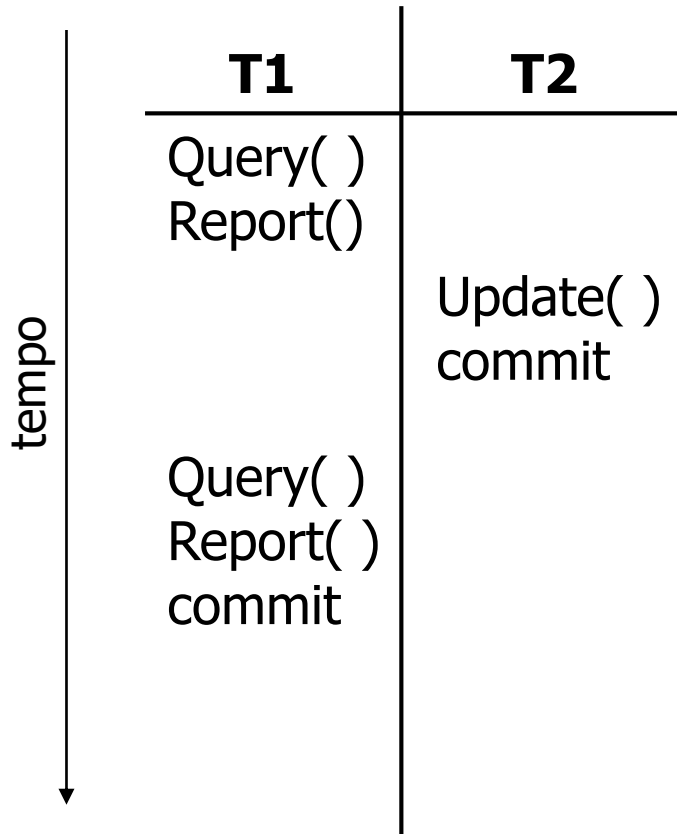
Problemas de Execução Intercalada

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

- Transação T lê um conjunto de tuplas que atendam a uma condição de consulta
- Transação T' **insere/remove/atualiza** uma tupla que atenderia a essa condição e é efetivada
- Se T refizer a mesma consulta, obterá um conjunto diferente de tuplas (*phantom read*)

Problemas de Execução Intercalada

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



Exemplo:

- **Transação T_1** : faz uma consulta que retorna a média geral dos alunos que têm média ponderada acima de 5.0, e gera um relatório
 - **Transação T_2** : atualiza as notas de alguns alunos e é efetivada
 - **T_1** refaz a consulta para gerar relatório com nro de alunos por faixa de média
- ⇒ **relatórios inconsistentes.**

Problemas de Execução Intercalada

- *Repeatable read vs Phantom read*
 - *Repeatable read*: lê valores diferentes de **um** mesmo dado que ainda está lá, mas foi alterado
 - *Phantom read*: lê **conjuntos** de dados diferentes, sendo que um dos conjuntos possui dados que não existem no(s) outro(s) conjunto(s) – fantasmas.

Problemas de Execução Intercalada: Isolamento

- Ocorrência de anomalias:

1. **Leitura inválida:** leu um dado errado, o que causou uma operação inconsistente;

- Um mesmo dado é escrito e lido em paralelo.

- **Solução:** bloquear o dado que foi escrito, mas que ainda não foi consolidado, até que a transação de escrita termine.

- Uma transação só vê apenas os dados committed antes de seu início.

Problemas de Execução Intercalada: Isolamento

- Ocorrência de anomalias:

2. **Leitura não repetível:** tentou ler um dado que foi alterado, impedindo que uma operação consistente fosse concluída;

- Um mesmo dado lido e escrito em paralelo.
- **Solução:** bloquear o dado que tenha sido lido até que a transação de leitura termine.
- Evita leitura inválida e, além disso, bloqueia os dados depois de sua primeira operação de leitura ou escrita.

Problemas de Execução Intercalada: Isolamento

- Ocorrência de anomalias:

3. **Leitura fantasma:** teve alterado o conjunto de tuplas envolvidas em uma seleção – fazendo com que uma operação tenha resultados diferentes em momentos diferentes da transação

- Não há concorrência por um mesmo dado específico – o que ocorre é que os dados envolvidos variam de maneira imprevisível.
- **Solução:** bloquear conjuntos de tuplas inteiros até que a transação termine.
- Evita leitura não repetível e, além disso, outras transações são impedidas caso elas suas operações intercalem com os dados da transação sendo feita.

Transações – Propriedades ACID

- **Atomicidade**: todas as operações de uma transação devem ser efetivadas; ou, na ocorrência de uma falha, nada deve ser efetivado
 - “**tudo ou nada**” – não se admite parte de uma operação
- **Consistência**: transações preservam a consistência da base
 - Estado inicial consistente \Rightarrow Estado final consistente
- **Isolamento**: a maneira como várias transações em paralelo interagem (o que pode ser lido e o que pode ser escrito por cada uma) deve ser bem definido
- **Durabilidade**: uma vez consolidada (committed) a transação, suas alterações permanecem no banco até que outras transações aconteçam

Problemas de Execução Intercalada: Isolamento

- Ocorrência de anomalias
 1. leitura inválida
 2. leitura não repetível
 3. leitura fantasma
- Solução via isolamento em diferentes graus
 - Read uncommitted
 - Read committed
 - Repeatable read
 - Serializable

Interpretação

Nível de isolamento	Anomalias que PODEM ocorrer		
	1) Leitura inválida	2) Leitura não repetível	3) Leitura fantasma
Leitura mesmo do que NÃO FOI committed	Sim	Sim	Sim
Leitura apenas do que FOI committed	Não	Sim	Sim
Leitura apenas se a leitura repetida for garantida	Não	Não	Sim
Torna a execução equivalente à execução em série	Não	Não	Não

Níveis de Isolamento em SQL99

Nível de isolamento	Anomalias que PODEM ocorrer		
	1) Leitura inválida	2) Leitura não repetível	3) 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

T1

1.1)Read(X)
1.2)Read(Y)
1.3)Write(X)
1.4)commit

T2

2.1)Read(X)
2.2)Read(Y)
2.3)Write(X)
2.4)Write(y)
2.5)commit

- 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.

Exercício

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

T1

1.1)Read(X)
1.2)Read(Y)
1.3)Write(X)
1.4)commit

T2

2.1)Read(X)
2.2)Read(Y)
2.3)Write(X)
2.4)Write(y)
2.5)commit

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

R.: 2.1,2.2,2.3,**1.1**,1.2,1.3,2.4,1.4,2.5

Aqui, T2 leu o valor de X e escreveu o valor de X (2.3), e antes de consolidar este valor, T1 leu o valor de X (**dirty read em 1.1**), e o reescreveu em 1.3. Em seguida T1 consolida os dados (1.4), seguido da consolidação dos dados por T2 (2.5). Qual é o valor de X, o que foi calculado por T2, ou o que foi calculado por T1?

Exercício

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

T1

1.1)Read(X)
1.2)Read(Y)
1.3)Write(X)
1.4)commit

T2

2.1)Read(X)
2.2)Read(Y)
2.3)Write(X)
2.4)Write(y)
2.5)commit

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

R.: exemplo, em T2 alguém transfere 100 de X para Y, situação em que X deve ser $X-100$. antes que T2 termine no entanto, alguém lê o saldo de $X-100$ e deposita mais 100. quando T1 terminar, T1 irá consolidar $(X-100)+100$, ao passo que T2 irá consolidar $X-100$, anulando o que foi feito por T1.

Exercício

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

T1

1.1)Read(X)
1.2)Read(Y)
1.3)Write(X)
1.4)commit

T2

2.1)Read(X)
2.2)Read(Y)
2.3)Write(X)
2.4)Write(y)
2.5)commit

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

R.: 1.1, 2.1, 2.2, 2.3, 1.2, 1.3, 1.4, 2.4, 2.5

Aqui T1 lê o valor de X, o que em seguida é alterado por T2 (2.3). T1 então tenta atualizar o valor de X (1.3), o qual agora não aceita a atualização inicialmente pretendida por T1.

Transações em Oracle

Transação em ORACLE

- Comando **SET TRANSACTION**

SET TRANSACTION

READ ONLY | READ WRITE | ISOLATION LEVEL
{SERIALIZABLE | READ COMMITTED} |

USE ROLLBACK SEGMENT rollback_segment

NAME 'nome_da_transacao';

Níveis de Isolamento de SQL99 no ORACLE

Nível de isolamento	Anomalias que podem ocorrer		
	Leitura inválida	Leitura não repetível	Leitura fantasma
Read uncommitted	Nunca permitido em Oracle.		
Read committed (padrão)	Não	Sim	Sim
Repeatable read	Não suportado especificamente (abrangido por serializable).		
Serializable	Não	Não	Não

Transações em ORACLE

- Transações explícitas iniciam com a cláusula SET TRANSACTION
- E terminam:
 - 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**)
- Execução de comando DDL

Transações em ORACLE

- Possuem quatro possibilidades:
 - **Modo de leitura:** para transações apenas leitura
 - READ-ONLY, e READ-WRITE
 - **Modo de isolamento:** para transações com atualizações
 - READ committed, e SERIALIZABLE

Transação em ORACLE

- Comando **COMMIT**
 - termina a transação
 - torna permanente as ações da transação
 - libera os recursos bloqueados

Transações em ORACLE

- Modo de isolamento: para transações com atualizações
 - READ committed (padrão):
 - Antes de uma operação, a transação aguarda até que quaisquer tuplas sendo atualizadas sejam liberadas e prossegue
 - A transação “vê” apenas dados consolidados (committed) antes do início de uma dada **operação**
 - Idéia de Snapshot dos dados antes de uma operação – vários snapshots do banco.

Transações em ORACLE

- Modo de isolamento: para transações com atualizações
 - SERIALIZABLE:
 - Caso uma tupla seja alterada após o início da transação serializable, se houver uma tentativa de alteração desta tupla, será jogada a exceção:
ORA-08177: Can't serialize access for this transaction.
 - O Oracle informa que não é capaz de tornar a concorrência semelhante a um processamento em série
 - A transação “vê” apenas dados modificados pela própria transação e dados efetivados antes do início da transação
 - Idéia de Snapshot dos dados antes de uma transação inteira – um único snapshot.