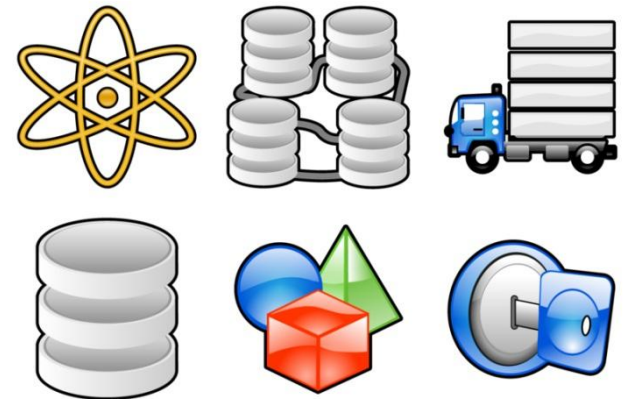


Banco de dados

Módulo 02 – Transações, Concorrência e Recuperação

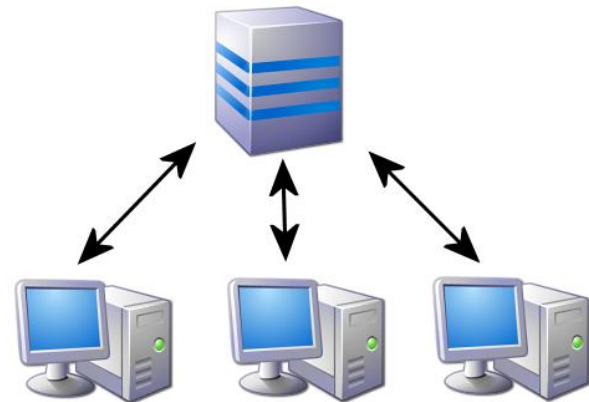
Curso Preparatório - ITnerante

Prof. Thiago Cavalcanti

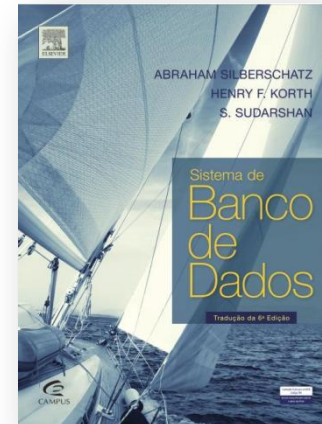
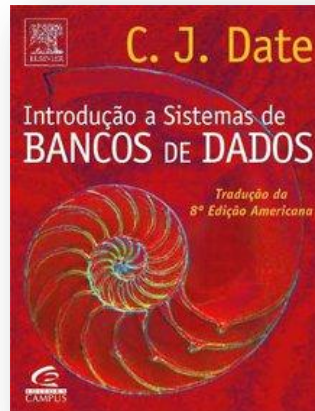
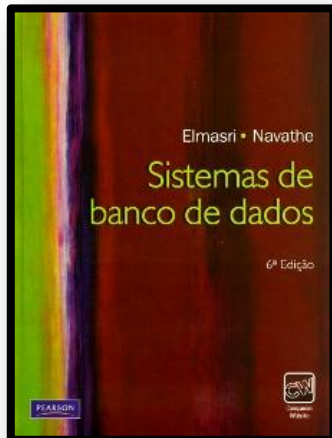


Ementa – Banco de dados

- Transações
- Controle de concorrência
- Recuperação após falha



Bibliografia

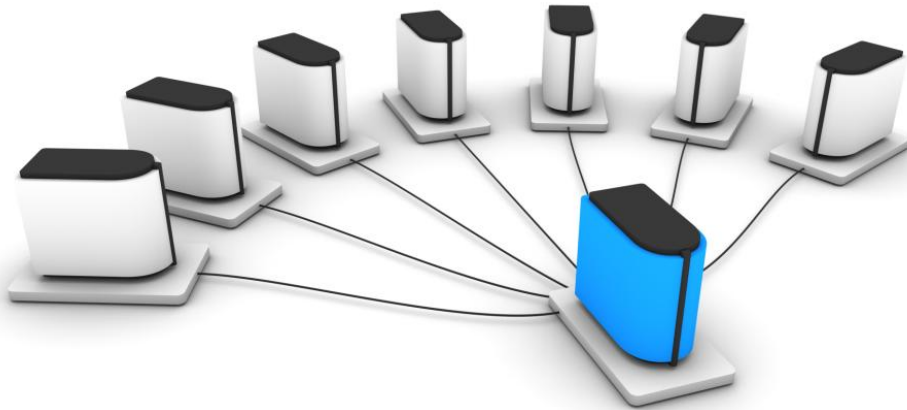


Objetivos

- Descrever os conceitos necessários aos sistemas de processamento de transações.
- Discutiremos as técnicas de controle de concorrência usadas para assegurar a propriedade de não interferência ou isolamento das transações executadas concorrentemente.
- Resumir algumas técnicas que podem ser usadas para recuperação (ou restauração) de falhas em sistemas de banco de dados.

Tire suas dúvidas

- rcthiago@gmail.com
- www.itnerante.com.br
- Lista: timasters@yahoogroups.com.br



Transações

GERENCIAMENTO DE TRANSAÇÕES

Conceito de transação



- É um programa em execução ou processo que inclui um ou mais acessos ao banco de dados, que efetuam leitura ou atualizações de seus registros.
- É uma unidade atômica de trabalho que estará completa ou não foi realizada.

Propriedades da uma transação



Atomicidade



Consistência



Isolamento



Durabilidade

Questão 1. CESPE - 2010 - MPU - Técnico de Informática

Julgue os próximos itens, que tratam de transações em sistemas de bancos de dados.

[107] Uma transação é uma unidade lógica de trabalho que tem como uma de suas propriedades a atomicidade. Segundo essa propriedade, uma vez que a transação tenha sido executada suas atualizações tornam-se permanentes no banco de dados.

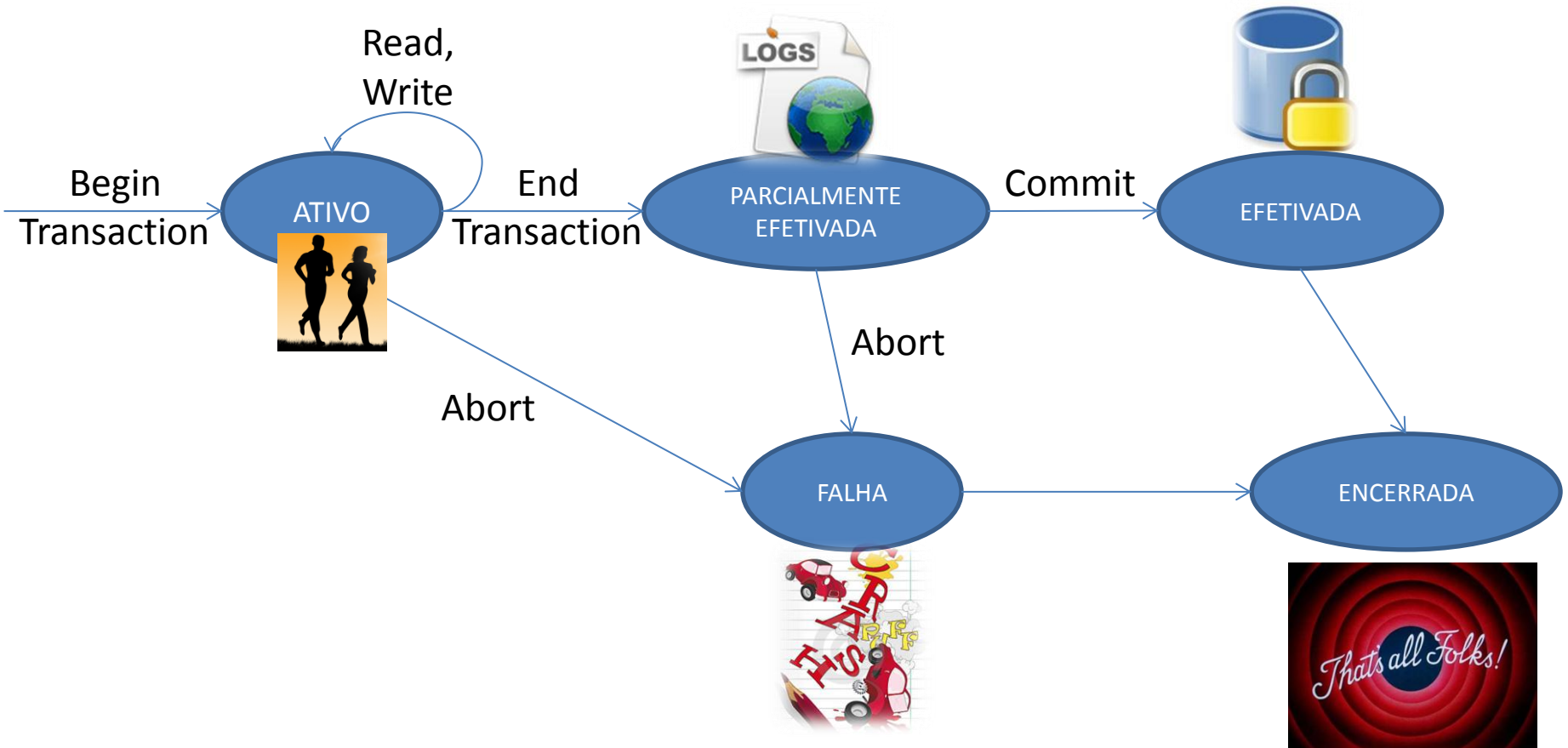
Operações

- **Begin_Transaction:** início da execução da transação
- **Read/Write:** especifica operações de leitura ou escrita de itens do BD que são executadas como parte da transação → (acesso ao banco)
- **End_Transaction:** especifica que as operações terminaram e marca o limite final da transação
- **Commit_Transaction:** sinaliza o final com sucesso de uma transação de modo que qualquer atualização executada pela transação possa ser efetivada no BD
- **Rollback (abort):** sinaliza que a transação terminou sem sucesso e o que foi executado deve ser desfeito

Estados de uma transação

- **Ativa:** poderá emitir operações de read e write.
- **Parcialmente efetivada:** alguns protocolos de restauração precisam garantir que uma falha de sistema não impossibilite a gravação permanente dos dados. → (gravada no logs)
- **Efetivada(*committed state*):** uma vez atendida todas as verificações
- **Falha:** se uma das verificações falhar ou se a transação for interrompida em seu estado ativo. A transação deverá ser revertida para desfazer os efeitos das operações de *WRITE* no BD.
- **Encerrada:** Quando a transação deixa o sistema.

Fluxo de uma transação



O Log do Sistema

- Utilizado para permitir recuperação de falhas de transações.
- Também chamado de Journal
 - Registra todas as operações que afetam valores de itens do BD
 - Guardado em disco
 - Periodicamente sofre backup em fita

Tipos de entradas no Log

- [start_transaction, T]
- [write_item, T, X, old-value, new-value]
- [read_item, T, X]
- [commit, T]
- [abort, T]



Comentário: Log de Redo e Log de Auditoria

Operações Adicionais

- **Undo (desfazer):** similar ao Rollback mas aplicada a uma única operação



- **Redo (refazer):** especifica que certas operações de uma transação têm que ser refeitas para garantir que todas as operações de uma transação efetivada (committed) foram aplicadas com sucesso ao BD



Ponto de Efetivação

- Uma transação T alcança seu ponto de efetivação (*commit point*) quando
 - Todas as suas operações que acessam o banco de dados estão sendo executadas com sucesso e;
 - O efeito de todas elas estiverem sendo gravados no log.
- Após o ponto de efetivação, a transação é dita **efetivada** e seu efeito será gravado de modo permanente no BD.
- Gravação forçada (force-writting)

Gerenciamento de Transações

- **Commit Point**, já falamos sobre
 - Quando todas as operações que acessam o BD foram executadas com sucesso e o efeito de todas as operações da transação no BD já **foram gravadas no log**
- **Checkpoints no Log do Sistema**
 - Escritos no log periodicamente quando **o sistema grava no disco** todas as operações **WRITE** de transações efetivadas

Checkpoints no Log do Sistema

- O gerenciamento de recuperação do SGBD deve decidir quais os intervalos em que devem ocorrer checkpoints
 - Em unidade de tempo
 - Em número de transações efetivadas após o último checkpoint



Ações de um checkpoint

- **Suspende** temporariamente a **execução** de transações
- **Forçar a escrita** de todas as **operações** de modificação das transações efetivadas do buffer **no disco**
- **Escrever** um registro [**checkpoint**] no log e forçar a escrita do log no disco
- Voltar à execução de transações

Escalonamento ou História de Transações

- A ordem de execução das operações de várias transações executando concorrentemente na forma intercalada é conhecida como:

–Plano de execução

- Baseados na Restaurabilidade
- Baseados em Serialidade

T ₁	T ₂
<pre>read (A); A := A - 50; write (A); read (B); B := B + 50; write (B);</pre>	<pre>read (A); temp := A * 0.1; A := A - temp; write (A); read (B); B := B + temp; write (B);</pre>

Plano de Execução de Transações

- Definição formal:
 - Um plano (ou histórico) S de n transações T_1, T_2, \dots, T_n é a **ordenação das operações das transações** sujeitas a uma restrição
 - Para cada T_i que participe de S , **as operações de T_i em S deverão aparecer na mesma ordem** em que elas ocorrem em T_i .

Operações em conflito

- Duas operações são ditas em conflito se elas satisfazem todas as três condições seguintes:
 1. Pertencerem a transações diferentes
 2. Acessarem o mesmo item X
 3. Pelo menos uma das operações ser um escrever_item(X).

Plano Completo

- Um plano S , com n transações T_1, T_2, \dots, T_n é chamado um plano completo se forem garantidas as seguintes condições:
 1. As operações em S são exatamente as operações de T_1, T_2, \dots, T_n , tendo **um commit ou um abort como última operação** de cada transação no plano.
 2. Para quaisquer **pares de operações** da mesma operação T_i , sua **ordem de aparecimento** em S será a **mesma** de T_i
 3. Para quaisquer **duas operações conflitantes**, uma das duas precisa aparecer antes da outra no plano.

Planos restauráveis e não-restauráveis

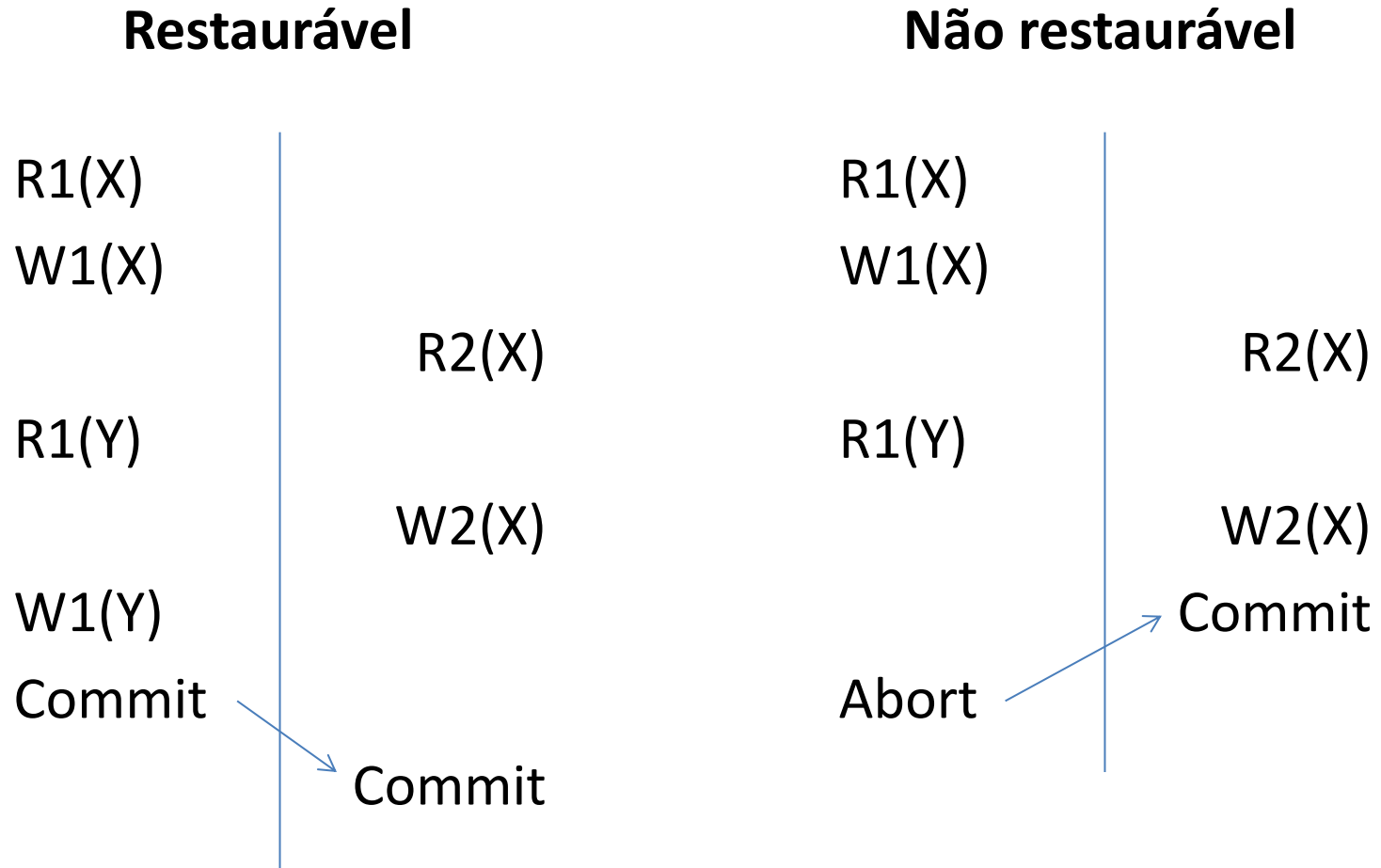
- **Restauráveis**

- É aquele na qual, para cada par de transações T_i e T_j , tal que T_j leia itens de dados previamente escritos por T_i , a operação de efetivação de T_i apareça antes da operação de efetivação de T_j .



Uma vez efetivada uma transação T , nunca mais seja necessário revertê-la.

Vamos a um exemplo



Reversão em cascata

- Uma transação não efetivada tem de ser revertida porque leu um item de um transação que falhou.
 - S: $r_1(X)$; $w_1(X)$; $r_2(X)$; $r_1(Y)$; $w_2(X)$; $w_1(Y)$; a_1 ; a_2 ;
- É importante estabelecer planos livres de reversão em cascata.
 - Solução: Uma transação somente ler os itens que foram gravados por transações efetivadas.

Plano estrito (strict schedule)

- As transações não poderão nem ler nem gravar um item X até a ultima transação que grave X tenha sido efetivada.
- Simplificam a restauração
 - Desfazer uma operação de escrita de uma transação é simplesmente restaurar a imagem anterior.

Serialidade (Serializability)

- **Planos seriais**
 - As operações de cada transação são executadas **consecutivamente**, sem intercalação das operações de outra transação.
 - Apenas uma transação ativa por vez
 - Limitam a concorrência ou intercalação de transações.
 - São inaceitáveis na prática.
- **Planos não seriais**

O conceito de **serialidade** de planos é usado para identificar quais **planos** são **corretos** quando há intercalação das operações das transações na execução dos planos

Equivalência de planos

- Dois planos são chamados **resultado** **equivalentes** se produzirem o mesmo estado final do banco de dados.
- Para dois planos serem **equivalentes**, as operações aplicadas a cada item de dado afetado por eles devem aparecer **na mesma ordem** em ambos os planos.
 - Equivalência de conflito
 - Se a ordem das operações conflitantes forem a mesma.
 - Equivalência de visão (veremos já já!)

Planos Conflito serializáveis

- Uma plano S com n transações é dito serializável se ele for equivalente a algum plano serial com as mesmas n transações.
- Dizer que um plano S não-serial é serializável é o mesmo que dizer que ele é correto.



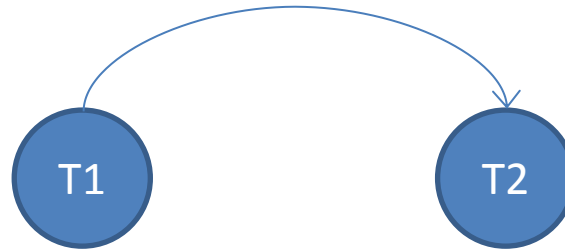
Questão 2. CESGRANRIO - 2008 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

[60] Uma transação em um sistema de banco de dados pode ser descrita como uma unidade de execução de programa que acessa e, possivelmente, atualiza vários itens de dados. Para o gerenciamento de transações em sistemas de bancos de dados, assinale a afirmação correta.

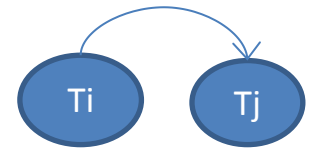
- (A) Atomicidade para uma transação significa que cada operação de uma transação é refletida na base de dados ou nenhuma operação o é.
- (B) Um escalonamento é a ordem de execução de várias transações de forma concorrente, mas nunca entrelaçada.
- (C) Integridade para uma transação significa que a execução de uma transação de forma isolada (sem sofrer concorrência de outra transação) preserva a consistência da base de dados.
- (D) Mesmo após a execução de um comando commit, é possível a uma transação retornar a seu estado inicial com a utilização de rollback.
- (E) Em transações que ocorrem simultaneamente, não existe qualquer garantia de que os dados lidos em um determinado momento são consistentes, independente do nível de isolamento utilizado

Testando o conflito serialidade

- Grafo de precedência



- Uma seta será criada caso alguma das operações de T_j apareça no plano antes de alguma operação conflitante de T_k .



1. Para cada transação T_i participante do plano S , **criar um nó rotulado T_i** no grafo de precedência
2. Para cada caso em S em que T_j executar um **ler_item(X)** depois que uma T_i executar um **escrever_item(X)**, criar uma seta $T_i \rightarrow T_j$ no grafo de precedência.
3. Para cada caso em S em que T_j executar um **escrever_item(X)** depois que uma T_i executar um **ler_item(X)**, criar uma seta $T_i \rightarrow T_j$ no grafo de precedência.
4. Para cada caso em S em que T_j executar um **escrever_item(X)** depois que uma T_i executar um **escrever_item(X)**, criar uma seta $T_i \rightarrow T_j$ no grafo de precedência.
5. O plano S será serializável se, o grafo de precedência **não contiver ciclos**.

(a)

Transação T_1	Transação T_2	Transação T_3
read_item(X); write_item(X); read_item(Y); write_item(Y);	read_item(Z); read_item(Y); write_item(Y); read_item(X); write_item(X);	read_item(Y); read_item(Z); write_item(Y); write_item(Z);

(b)

Tempo ↓

Transação T_1	Transação T_2	Transação T_3
read_item(X); write_item(X);	read_item(Z); read_item(Y); write_item(Y);	read_item(Y); read_item(Z);
read_item(Y); write_item(Y);	read_item(X);	write_item(Y); write_item(Z);
	write_item(X);	

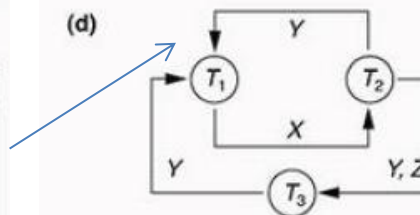
Schedule E

(c)

Tempo ↓

Transação T_1	Transação T_2	Transação T_3
read_item(X); write_item(X);		read_item(Y); read_item(Z);
read_item(Y); write_item(Y);	read_item(Z);	write_item(Y); write_item(Z);
	read_item(Y); write_item(Y); read_item(X); write_item(X);	

Schedule F



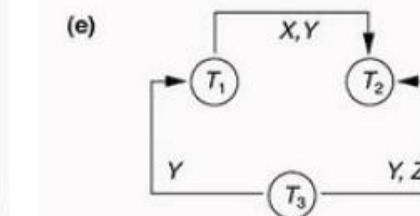
Schedules seriais equivalentes

Nenhum

Motivo

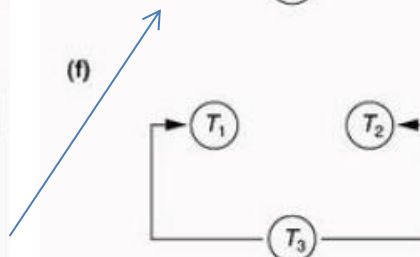
Ciclo $X(T_1 \rightarrow T_2), Y(T_2 \rightarrow T_1)$

Ciclo $X(T_1 \rightarrow T_2), YZ(T_2 \rightarrow T_3), Y(T_3 \rightarrow T_1)$



Schedules seriais equivalentes

$T_3 \rightarrow T_1 \rightarrow T_2$



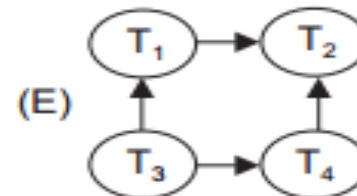
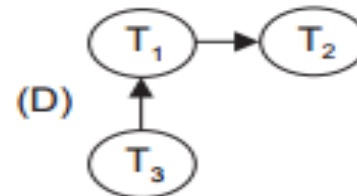
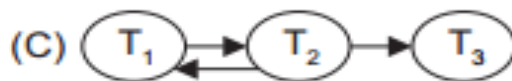
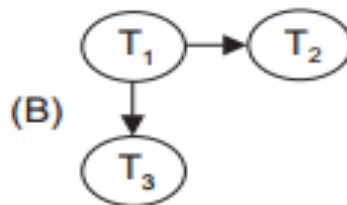
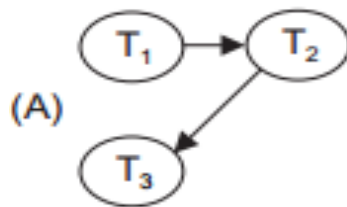
Schedules seriais equivalentes

$T_3 \rightarrow T_1 \rightarrow T_2$

$T_3 \rightarrow T_2 \rightarrow T_1$

Questão 3. CESGRANRIO - 2008 - BNDES - Profissional Básico - Especialidade - A de Sistemas – Suporte

- [66] No âmbito de banco de dados, que grafo de precedência representa um escalonamento que NÃO é serializável quanto ao conflito?



Aplicação da serialidade

- A abordagem adotada na maioria dos SGBDs comerciais é definir **protocolos** (conjunto de regras) que:
 - Devem ser seguidos por **todas as transações individualmente**, ou impostos por um subsistema de controle de concorrência do SGBD
 - E garantirão a serialidade de todos os planos dos quais as transações participam.

Equivalência de visão e visão serialidade

- Diz que dois planos S e S' são **visão equivalente** se:
 1. O **conjunto de transações** participantes em S e S' seja o mesmo e que S e S' contenham as mesmas **operações** dessas transações.
 2. Para toda operação $ri(X)$ de T_i em S , se o valor X lido pela operação tiver sido alterado por uma operação $wj(X)$ de T_j , a mesma condição deverá ser garantida para o valor X lido pela operação $ri(X)$ de T_i em S' .
 3. Se a operação $wk(Y)$ de T_k for a última operação a gravar o item Y em S , então $wk(Y)$ de T_k também deverá ser a última operação a gravar Y em S' .

SUPOORTE A TRANSAÇÃO EM SQL

- Transação
 - Uma unidade lógica de trabalho
 - Garantida como atômica
- Em SQL não há BEGIN TRANSACTION explícito
- ROLLACK e COMMIT são explícitos
- Características específicas:
 - Modo de acesso
 - Tamanho da área de diagnóstico
 - Nível de isolamento

Classificações

- Modo de acesso
 - READ ONLY
 - READ WRITE
- Tamanho da área de diagnostico
 - DIAGNOSTIC SIZE n
- Nível de isolamento (ISOLATION LEVEL <I>)
 - READ UNCOMMITTED (leitura suja)
 - READ COMMITED (leitura não-repetível)
 - REPEATABLE READ (fantasmas)
 - SERIALIZABLE

Dirty read (Leitura Suja)



Transaction 1

```
/* Query 1 */  
SELECT age FROM users WHERE id = 1;  
/* will read 20 */
```

Transaction 2

```
/* Query 2 */  
UPDATE users SET age = 21 WHERE id = 1;  
/* No commit here */
```

```
/* Query 1 */  
SELECT age FROM users WHERE id = 1;  
/* will read 21 */
```

```
ROLLBACK; /* lock-based DIRTY READ */
```


Non-repeatable read (Leitura não-repetível)

Transaction 1

```
/* Query 1 */  
SELECT * FROM users WHERE id = 1;
```

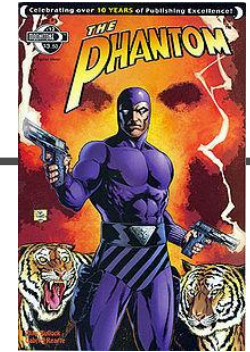
```
/* Query 1 */  
SELECT * FROM users WHERE id = 1;  
COMMIT; /* lock-based REPEATABLE READ */
```

Transaction 2

```
/* Query 2 */  
UPDATE users SET age = 21 WHERE id = 1;  
COMMIT; /* in multiversion concurrency  
control, or lock-based READ COMMITTED */
```



Phantom read (fantasmas)



Transaction 1

```
/* Query 1 */  
SELECT * FROM users  
WHERE age BETWEEN 10 AND 30;
```

Transaction 2

```
/* Query 2 */  
INSERT INTO users VALUES ( 3, 'Bob', 27 );  
COMMIT;
```

```
/* Query 1 */  
SELECT * FROM users  
WHERE age BETWEEN 10 AND 30;
```

Exemplo

```
EXEC SQL WHENEVER SQLERROR GOTO UNDO;  
EXEC SQL SET TRANSACTION
```

```
    { READ WRITE  
      DIAGNOSTIC SIZE 5  
      ISOLATION LEVEL SERIALIZABLE }  
    }
```

```
EXEC SQL INSERT INTO EMPREGADO (PNOME, UNOME, SSN,  
DNO, SALARIO) VALUES ('Thiago', 'Cavalcanti', 000457878, 2,  
12000);
```

```
EXEC SQL UPDATE EMPREGADO
```

```
    SET SALARIO = SALARIO * 1,05 WHERE DNO = 2;
```

```
EXEC COMMIT;
```

```
GOTO THE_END;
```

```
UNDO: EXEC SQL ROLLBACK;
```

```
THE_END: ...;
```

Questão 4. CESGRANRIO - 2010 - EPE - Analista de Gestão Corporativa - Tecnologia da Informação

[22] Um administrador de banco de dados recebeu a solicitação de atualizar, no sistema de controle de estoque, o nome de um produto. De acordo com esse pedido, ele deveria alterar, diretamente no banco de dados, o valor do campo nome em uma única linha de uma tabela. No entanto, ele errou ao colocar o nome solicitado no update. Ao perceber o equívoco, aplicou o rollback nessa transação. Nesse período, um usuário reclamou que, ao realizar uma consulta no sistema, esse produto apareceu com o nome errado. Considerando-se que o erro percebido pelo usuário foi o mesmo introduzido pelo administrador, que fenômeno ocorreu nessa situação?

- (A) Dirty Read
- (B) Nonrepeatable Read
- (C) Phantom Read
- (D) Serializable Write
- (E) WriteNeverLock



CONTROLE DE CONCORRÊNCIA

Porque o controle de concorrência é necessário?

O problema da atualização perdida

O problema da atualização temporária

O problema do sumário incorreto



Controle de Concorrência

- **Definição:** Concorrência é a propriedade de uma transação poder ser **executada em paralelo** com outras transações
- **Justificativa de concorrência:**
 - Com a execução de várias transações ao mesmo tempo, o processador pode ser compartilhado entre essas transações, **melhorando a eficiência global** do computador dado que uma maior quantidade de trabalho é executada em menos tempo

Controle de Concorrência

- **Justificativa de um controle:**
 - É necessário que o sistema monitorize a interação entre transações concorrentes, de modo a evitar que elas destruam a consistência do BD e mantenham o isolamento das transações.
- **Técnicas ou Protocolos:**
 - Técnica de bloqueio
 - Timestamp
 - Protocolos com Base em Gráficos

Questão 5. CESPE - 2010 - MPU - Técnico de Informática

Julgue os próximos itens, que tratam de transações em sistemas de bancos de dados.

[106] Os mecanismos de controle de concorrência implementados em sistemas de bancos de dados visam garantir que as transações tenham a propriedade de isolamento.

Técnicas de Bloqueio (Locks)

- LOCK: Variável associada a um item de dados no BD que descreve o status desse item com respeito a possíveis operações a serem aplicadas a ele
 - Binário
 - Múltiplo

LOCK BINÁRIO

- Dois estados ou valores:
 - locked (1) ou
 - unlocked (0)
- Um lock distinto é associado a cada item do BD - referenciado como lock (x) para o item x
- Operações incluídas nas transações : lock_item e unlock_item, implementadas como operações indivisíveis
- Um gerenciador é mantido pelo SGBD para registrar e controlar o acesso a locks
- Registro de lock: <nome-do-item, LOCK>
- Tabela de locks: mantém esses registros

LOCK BINÁRIO (Regras)

- Uma transação T tem que executar uma operação `lock_item(x)` antes de qualquer `read_item` ou `write_item` executada por T
- Uma transação T tem que executar a operação `unlock_item(x)` após todo `read/write` completados em T
- Uma transação não pode executar outra operação `lock_item(x)` se já tem um lock sobre x
- Uma transação T não pode executar um `unlock_item(x)` a menos que tenha um lock sobre x
- Apenas uma transação pode ter um lock num dado item

LOCK MÚLTIPLO

- Três operações (indivisíveis)
 - `read_lock(x)` - lock compartilhado – `lock-S(x)`
 - `write_lock(x)` - lock exclusivo - `lock-X(x)`
 - `unlock(x)`
- Registro de Lock
 - `<nome-do-item, LOCK, (número-de-leituras)>`

	Compartilhado	Exclusivo
Compartilhado	Verdadeiro	Falso
Exclusivo	Falso	Falso

LOCK MÚLTIPLO (Regras)

- Uma transação T tem que executar uma operação `read_lock(x)` ou `write_lock(x)` antes de qualquer `read_item(x)` em T
- Uma transação T tem que executar uma operação `write_lock(x)` antes de qualquer `write_item(x)` em T
- Uma transação tem que executar uma operação `unlock(x)` após todas as operações `read_item(x)` e `write_item(x)` completadas em T

LOCK MÚLTIPLO (Regras)

- Uma transação T não executará um `read_lock(x)` se já tem um lock (read) compartilhado ou um lock (write) exclusivo em x (pode ser relaxado)
- Uma transação T não executará outro `write_lock(x)` se já tem um lock (write) exclusivo ou um lock (read) compartilhado em x (pode ser incrementado)
- Uma transação T não executará um `unlock(x)` a menos que já tenha um lock compartilhado ou exclusivo em x

INCREMENTO/DECREMENTO DE LOCK

- INCREMENTO DE LOCK
 - Uma transação após ter um `read_lock(x)` e sendo a única que detém `x`, pode posteriormente executar um `write_lock(x)` sobre `x`
- DECREMENTO DE LOCK
 - Após ter um lock exclusivo sobre um item `x`, uma transação `T` pode decrementar o lock, executando um `read_lock(x)`
- BLOQUEIOS EXCLUSIVOS
 - Se uma transação `T` contiver um bloqueio exclusivo em algum objeto, então nenhuma transação distinta `T'` pode fazer bloqueio daquele objeto até que `T` libere o seu bloqueio

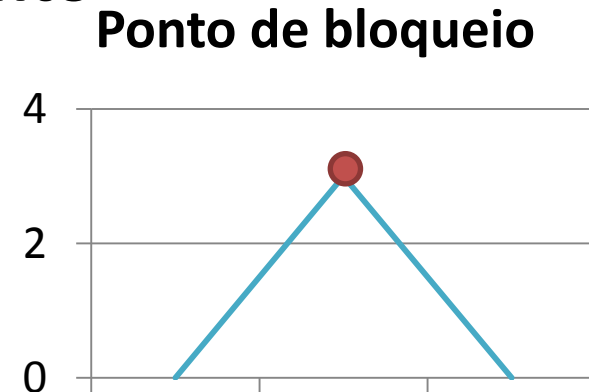
Como garantir a serializabilidade?

- Locks binários ou múltiplos não garantem a serializabilidade de escalonamentos nos quais as transações participam.
- É necessário um **protocolo adicional** que contempla o posicionamento de locks e unlocks em cada transação
 - Protocolo de bloqueio



Protocolo - BLOQUEIOS DE DUAS FASES

- Se todas as transações obedecerem às seguintes regras
 1. Antes de operar sobre qualquer objeto, a transação primeiro adquire um bloqueio sobre aquele objeto – fase de **expansão**
 2. Após liberar o bloqueio, a transação não adquire mais bloqueios – fase de **encolhimento**
- **Upgrade e downgrade**



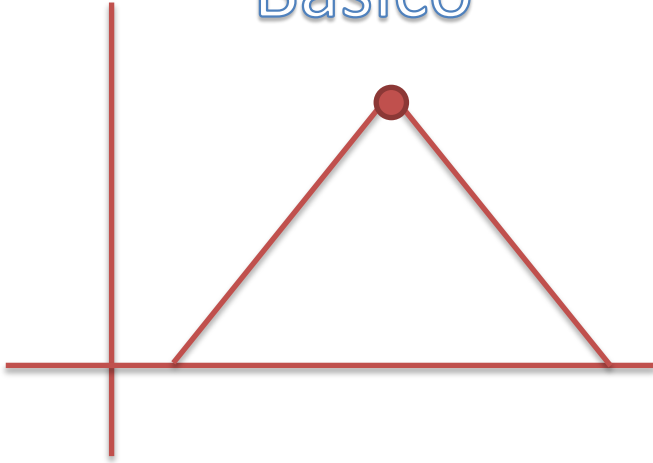
Variação do protocolo

- BÁSICO (visto anteriormente)
- CONSERVADOR OU ESTÁTICO:
 - Bloqueia todos os itens que acessará, antes de iniciar a execução da transação
 - Só bloqueia quando todos estão disponíveis.
- ESTRITO OU SEVERO (mais popular) :
 - Não libera os locks de escrita até o commit ou abort
- RIGOROSO
 - Não libera os locks de escrita e leitura até o commit ou abort
 - É mais fácil de implementar do que o estrito.

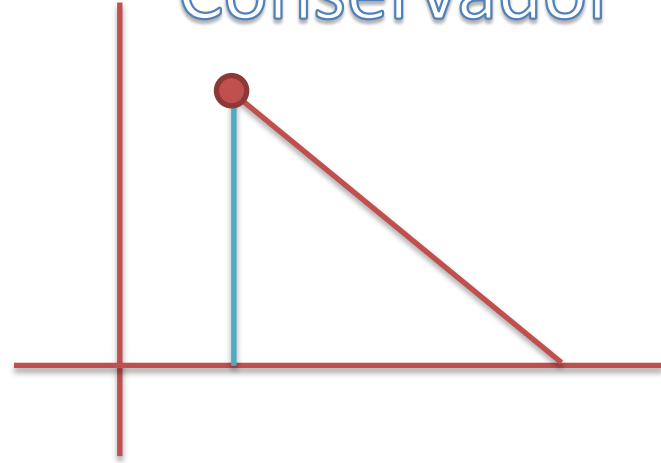
Podem Causar Deadlock ou Starvation

Veamos no Gráfico agora

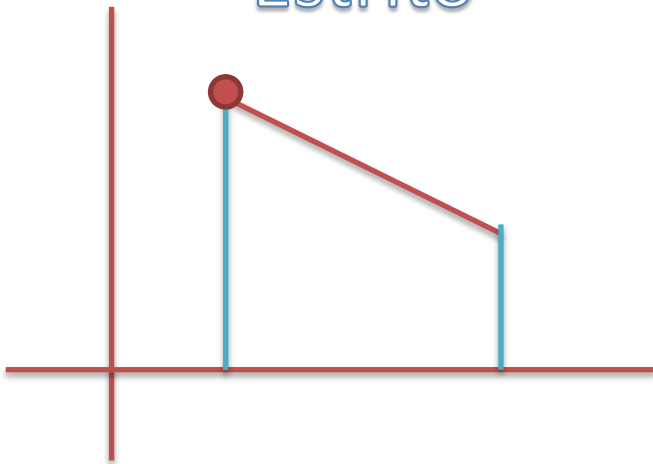
Básico



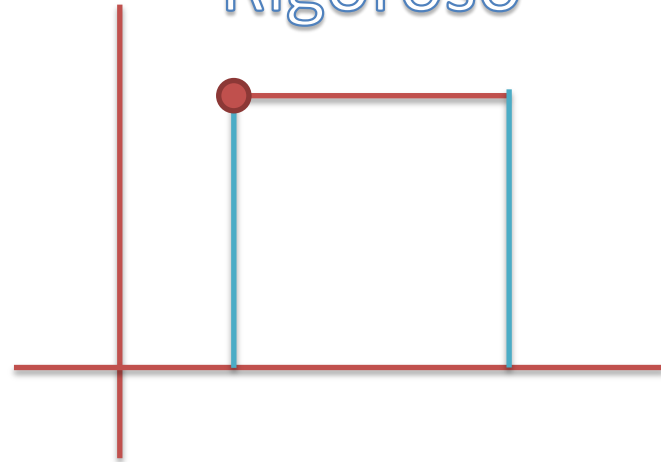
Conservador



Estrito



Rigoroso



Questão 6. CESPE - 2009 - TCU - Analista de Controle Externo - Tecnologia da Informação - Prova 2

T1	T2
lock(p1)	
read(p1)	
write(p1)	lock(p1)
lock(p2)	
unlock(p1)	
	read(p1)
	write(p1)
	lock(p2)
	unlock(p1)
read(p2)	
write(p2)	
unlock(p2)	
	read(p2)
	write(p2)
	unlock(p2)

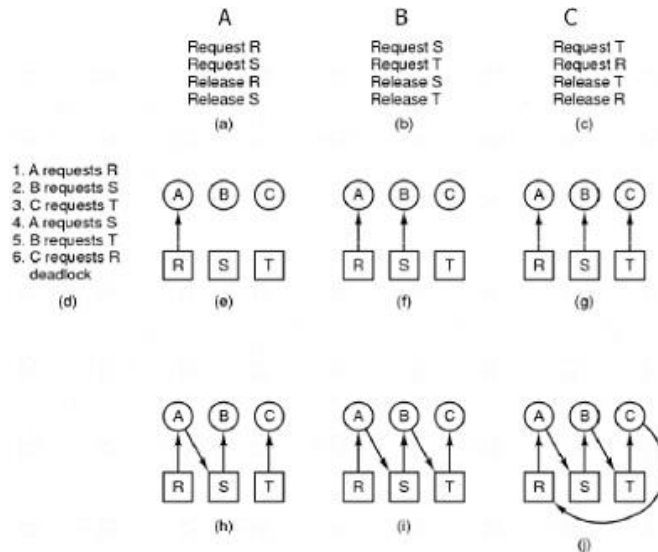
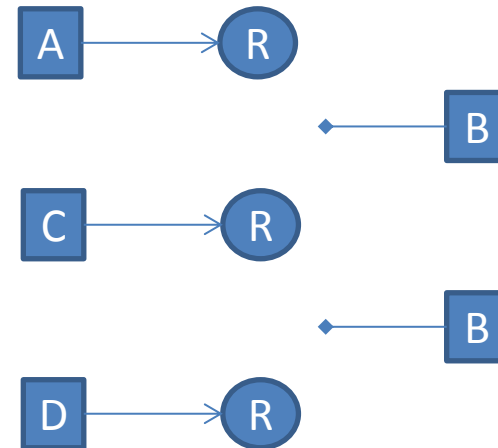
Considerando as transações T1 e T2 apresentadas acima, julgue o item subsequente.

[129] Na situação em questão, a transação T2 pode ver os writes incrementais de T1. Esse enfoque diminui a concorrência do sistema.

Deadlock e Starvation

T1	T2
Lock-X(B)	
Read(B)	
B := B - 50	
Write(B)	
	Lock-S(A)
	Read(A)
	Lock-S(B)
Lock-X(A)	

A, C, D ... Bloqueio compartilhado
B Bloqueio exclusivo -- Starvation



DEADLOCK

- É uma situação em que duas ou mais transações estão em estado simultâneo de espera, cada uma aguardando que uma das demais libere um bloqueio para ela poder prosseguir
- Principais métodos para solucionar o impasse (podem resultar na repetição da transação)

DEADLOCK

- Protocolo de Prevenção de Deadlock
 - Conservador: se algum dos itens não pode ser bloqueado, nenhum será bloqueado
 - Ordenado: tentar por uma ordenação em todos os itens e os locks só podem ocorrer segundo esta ordem



Protocolos

- Quando uma transação solicita um bloqueio de um registro que já está bloqueado por outra transação, então um dos dois procedimentos é seguido
- **Wait-die** - se a transação que solicitou o bloqueio é a mais antiga, pode aguardar. Se for a mais nova, sofre rollback e recomeça mais tarde com mesmo timestamping
- **Wound-wait** - se for a transação mais nova, pode aguardar. Se for a mais antiga, interrompe a mais nova, a qual sofre rollback. Recomeça mais tarde com mesmo timestamping

Vejam os de uma forma mais clara!

Wait-die



$$TS1(T_i) < TS(T_j)$$

(T_i)



$$TS1(T_i) > TS(T_j)$$

(T_i)



Wound-wait



$$TS1(T_i) < TS(T_j)$$

(T_i)



(T_j)



$$TS1(T_i) > TS(T_j)$$

(T_i)



Protocolo de Detecção e Recuperação

- Verificação da existência de deadlock
 - Sempre que uma solicitação de bloqueio cause “espera”
 - Permite detecção imediata do impasse
 - Acarreta um overhead
 - Em alguma base periódica
 - Reduz o overhead
 - Alguns deadlocks serão detectados tardiamente
 - Condições ideais
 - Pequena interferência entre as transações
 - Transações curtas
 - Locks de poucos itens
 - Carga da transação leve

Controle de Concorrência

- Quebra de deadlock consiste na escolha de uma das transações para forçá-la a um rollback
 - A que foi iniciada mais recentemente
 - A que tiver feito o menor número de bloqueios
 - A que tiver feito o menor número de atualizações
- Situação oposta
 - Usar um esquema de prevenção de deadlock

LIVELOCK

- A transação não pode prosseguir por um período indefinido de tempo enquanto outras transações continuam normalmente (esquema injusto de espera por lock)
 - Associar first-come-first-serve
 - Incremento de prioridade com o tempo

STARVATION

- Problema similar ao livelock, ocorre se o algoritmo seleciona a mesma transação como vítima repetidamente, causando abort repetidos e nunca acabando a execução
 - Os protocolos wait-die e wound-wait apresentados evitam starvation

Evitando inanição de transações

- Quando uma transação T_i solicita o bloqueio do item de dados Q de modo particular M , o bloqueio é concedido, contanto que:
 1. Não haja nenhuma outra transação com bloqueio sobre Q cujo modo de bloqueio seja conflitante com M .
 2. Não haja nenhuma outra transação que esteja esperando um bloqueio sobre Q e que tenha feito sua solicitação antes de T_i .

Questão 7. FCC - 2011 - TRE-RN - Analista Judiciário - Análise de Sistemas

[54]. No âmbito dos bancos de dados, quando cada transação em um conjunto de duas ou mais transações espera por algum item que esteja bloqueado por alguma outra transação no conjunto, o que ocorre é

- (A) commit.
- (B) timestamp.
- (C) deadlock.
- (D) rollback.
- (E) starvation.

Protocolo com base em Grafo (Graph-based)

- Exige conhecimento anterior sobre a ordem na qual os itens de banco de dados são acessados.
- Exemplo de protocolo: é permitida somente a instrução de bloqueio lock-X. Cada transação T_i , pode bloquear um item de dado no máximo uma vez e deve observar as seguintes regras:
 1. O primeiro bloqueio feito por T_i pode ser sobre qualquer dado
 2. Subsequentemente, um certo item de dado Q pode ser bloqueado por T_i somente se os pais de Q estiverem bloqueados por T_i
 3. Itens de dados podem ser desbloqueados a qualquer momento
 4. Um item de dado foi bloqueado e desbloqueado por T_i não pode ser rebloqueado por T_i .

Protocolo com base em Grafo (Graph-based)

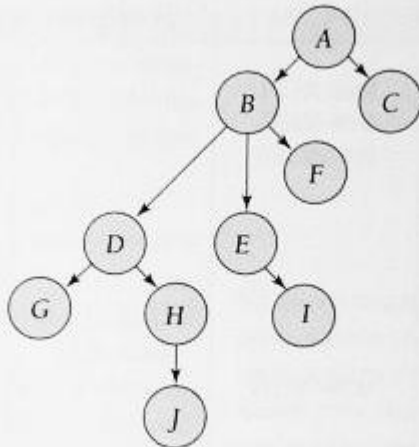
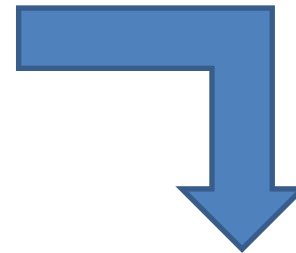


Figura 16.11 Gráfico de banco de dados estruturado em árvore.



T_{10}	T_{11}	T_{12}	T_{13}
lock-X(B)	lock-X(D) lock-X(H) unlock(D)		
lock-X(E) lock-X(D) unlock(B) unlock(E)		lock-X(B) lock-X(E)	
	unlock(H)		
lock-X(G) unlock(D)			lock-X(D) lock-X(H) unlock(D) unlock(H)
		unlock(E) unlock(B)	
unlock (G)			

Figura 16.12 Schedule serial sob o protocolo de árvore.

Timestamp

- A cada transação T_i do sistema associamos um único timestamp fixo, denotado por $TS(T_i)$
- Esse é criado pelo SBD antes que a transação T_i inicie sua execução
- Se uma transação T_i recebeu o $TS(T_i)$ e uma nova transação T_j entra no sistema, então
 - $TS(T_i) < TS(T_j)$
- Duas forma de implementar
 - Clock(relógio do sistema) ou contador lógico

O protocolo de ordenação por Timestamp

1. Suponha que a transação T_i emita uma instrução $\text{read}(Q)$
 - a) Se $TS(T_i) < W\text{-timestamp}(Q)$, então T_i precisa ler um valor de Q que já foi sobreposto. Assim, a operação read é rejeitada e T_i é desfeita.
 - b) Se $TS(T_i) \geq W\text{-timestamp}(Q)$, então a operação read é executada e $R\text{-timestamp}(Q)$ recebe o maior valor entre $R\text{-timestamp}(Q)$ e $TS(T_i)$
2. Suponha que a transação T_i emita uma $\text{write}(Q)$
 - a) Se $TS(T_i) < R\text{-timestamp}(Q)$, então o valor de Q que T_i está produzindo foi necessário antes e o sistema assumiu que aquele valor nunca seria produzido. Logo, a operação write é rejeitada e T_i é desfeita.
 - b) Se $TS(T_i) < W\text{-timestamp}(Q)$, então T_i está tentando escrever um valor obsoleto em Q . Logo, essa operação write é rejeitada e T_i é desfeita.
 - c) De outro modo, a operação write é executada e $W\text{-timestamp}(Q)$ é registrada em $TS(T_i)$

O protocolo de ordenação por Timestamp

- Uma transação T_i desfeita pelo esquema de controle de concorrência recebe um novo timestamp e é reiniciada.

$W\text{-TS}(A) = 0 \mid R\text{-TS}(A) = 0 \mid W\text{-TS}(B) = 0 \mid R\text{-TS}(B) = 0$

$TS(T1) = 1$

$TS(T2) = 2$

T1	T2
Read (B)	
	Read (B)
	$B := B - 50$
	Write(B)
Read(A)	
	Read(A)
Display (A+B)	
	$A = A + 50$
	Write (A)
	Display (A+B)

Técnicas Baseadas em Timestamping

- Identificador único assinalado a cada transação
- Ordenados segundo a ordem em que as transações começaram
- Principal vantagem: não usa bloqueios, logo deadlock é impossível
- Garante a serialização de conflito*

Regra de Escrita de Thomas

- Thomas' Write Rule
- Não impõe serialização de conflito, mas rejeita menos operações de escrita (write), modificando as verificações para escrever_item(X).
 1. Se $\text{read_TS}(X) > \text{TS}(T)$, então aborta e reverte T e rejeita a operação.
 2. Se $\text{write_TS}(X) > \text{TS}(T)$, então não executa a operação write, mas continua processando.*
 3. Se nem a condição (1) nem a condição (2) ocorrerem, então executa a operação de escrever_item(X) e aponta write_TS(X) para TS(T)
- Aumenta a concorrência em potencial

Travas

- Bloqueio de curta duração são chamados travas.
- As travas não seguem os protocolos usuais de controle de concorrência tal como o bloqueio de duas fases.
- Por exemplo, uma trava pode ser usadas para garantir a integridade física de uma página quando esta está sendo liberada do buffer para o disco.
- Uma trava seria fornecida para a página, a página escrita para o disco , então a trava seria liberada.

Questão 8. CESPE - 2009 - ANAC - Analista Administrativo - Tecnologia da Informação

[70] Para o SGBD viabilizar a execução de transações concomitantemente existem diversas técnicas de controle de concorrência que são utilizadas para garantir propriedade de não-interferência ou isolamento de transações. Uma dessas técnicas é o controle de concorrência baseado em ordenamento de registro de timestamp que utiliza o bloqueio combinado com o Protocol Two-Phase locking (2PL).

... Revisando!!

- Existem alguns protocolos de controle de concorrência em transações de banco de dados um desses é o protocolo de **BLOQUEIOS DE DUAS FASES** que faz com que todas as transações obedecem às seguintes regras:
 1. Antes de operar sobre qualquer objeto, a transação primeiro adquire um bloqueio sobre aquele objeto
 2. Após liberar o bloqueio, a transação não adquire mais bloqueios.
- Outro conjunto de protocolos de controle de concorrência utilizar **rótulos de tempo (timestamp)**.
 - Um timestamp é um identificador exclusivo para cada transação, gerado pelo sistema. Os valores de rótulo de tempo são gerados na mesma ordem que os tempos de início de transação.
- Vejam que são dois grupos de protocolos distintos (timestamp e bloqueio(em duas fases, por exemplo)).
 - A questão tenta combinar os dois, o que não faz sentido. Logo, alternativa está errada.



RECUPERAÇÃO APÓS FALHA (OU RESTAURAÇÃO)

Por que a Restauração é necessária?

1. O computador falhar
2. Um erro de transação ou sistema
3. Erros locais ou condições de exceção detectadas pela transação
4. Imposição do controle de concorrência
5. Falha de disco
6. Problemas físicos ou catástrofes



Recuperação após Falha (ou Restauração)

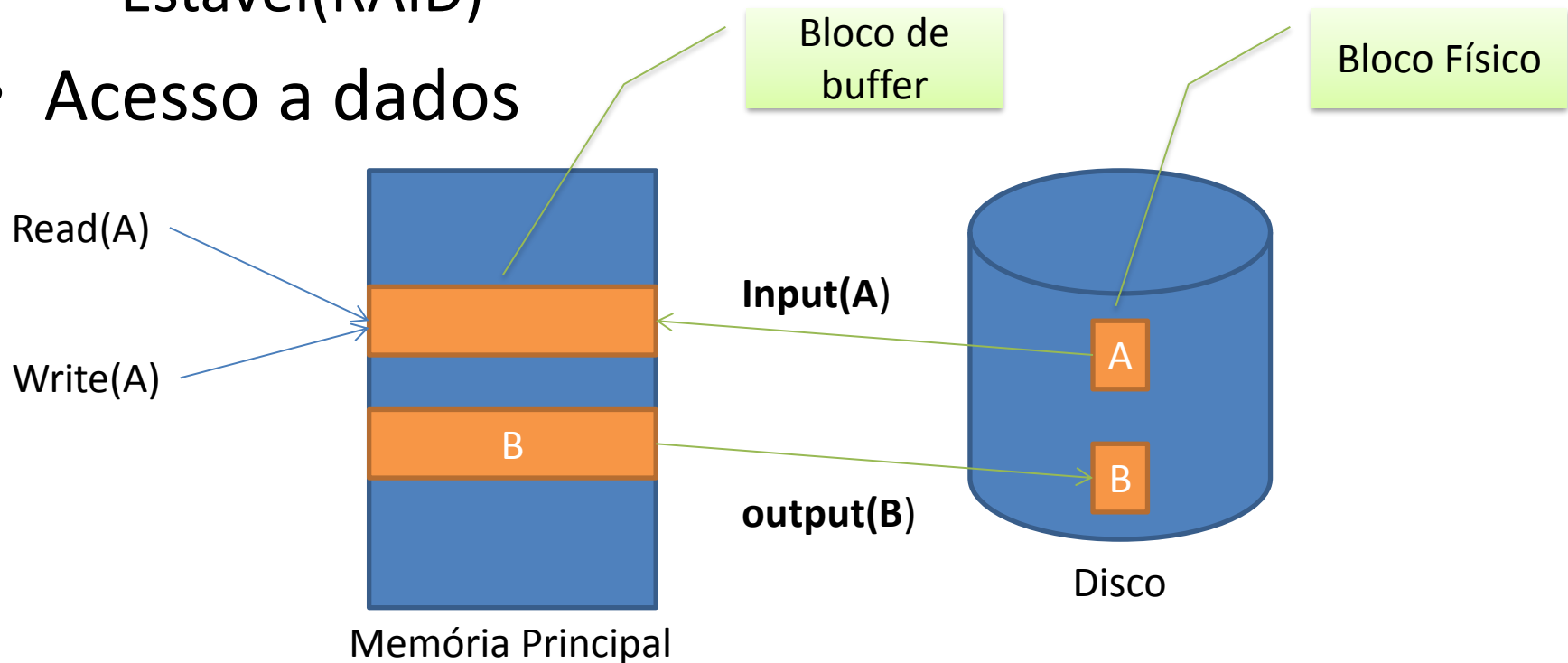
- Após uma falha na execução de uma transação, o BD deve ser restaurado para um estado consistente anterior
- O Sistema deve manter informações sobre as atualizações do BD em separado (LOG)

Classificação de Falha

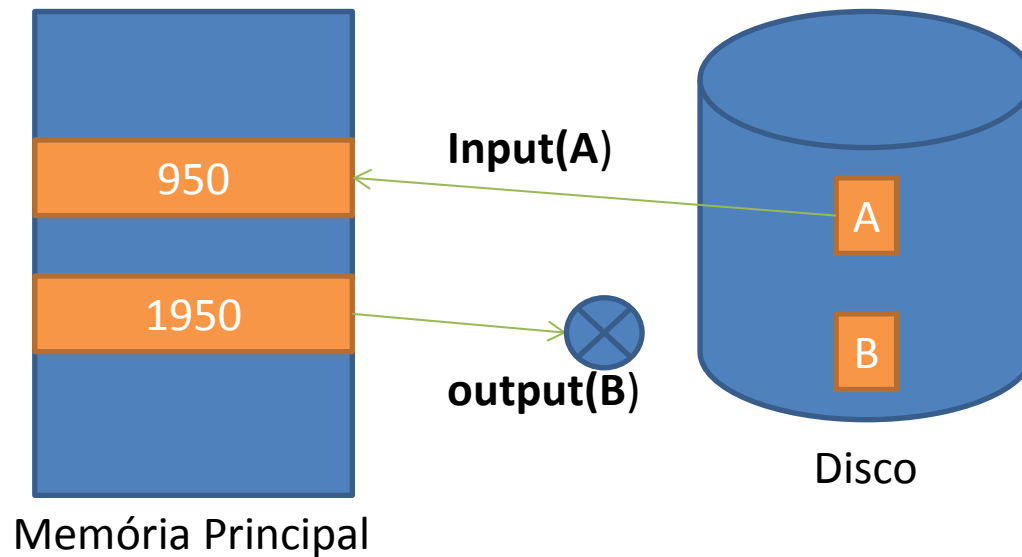
- Falha de transação
 - Erro lógico
 - Erro de sistema
- Queda do sistema
- Falha de disco

Estrutura de Armazenamento

- Tipos de armazenamento
 - Volátil, não-volátil (Queda no sistema)
 - Estável(RAID)
- Acesso a dados



Motivação ao uso de logs



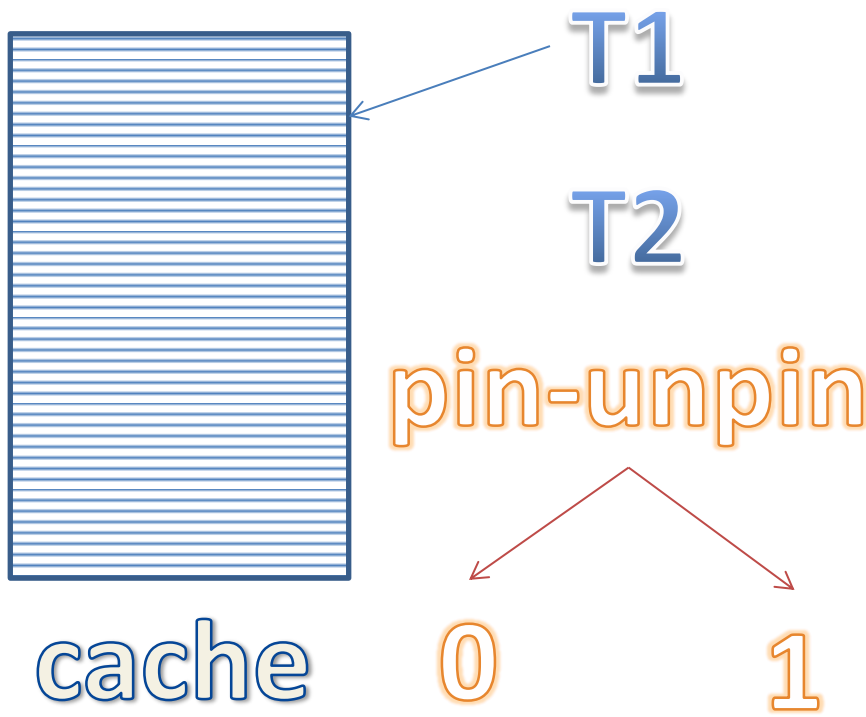
Recuperação baseada em Log

- Um registro de log descreve $\langle T_i, X_j, V_1, V_2 \rangle$:
 - Identificador da transação
 - Identificado do item de dado
 - Valor antigo
 - Valor novo
- $\langle T_i \text{ start} \rangle$, $\langle T_i \text{ commit} \rangle$, $\langle T_i \text{ abort} \rangle$
- Sempre que uma transação realiza uma escrita, é essencial que o registro de log para aquela escrita seja criado antes de o BD ser modificado.
- **Devem** residir em armazenamento estável.

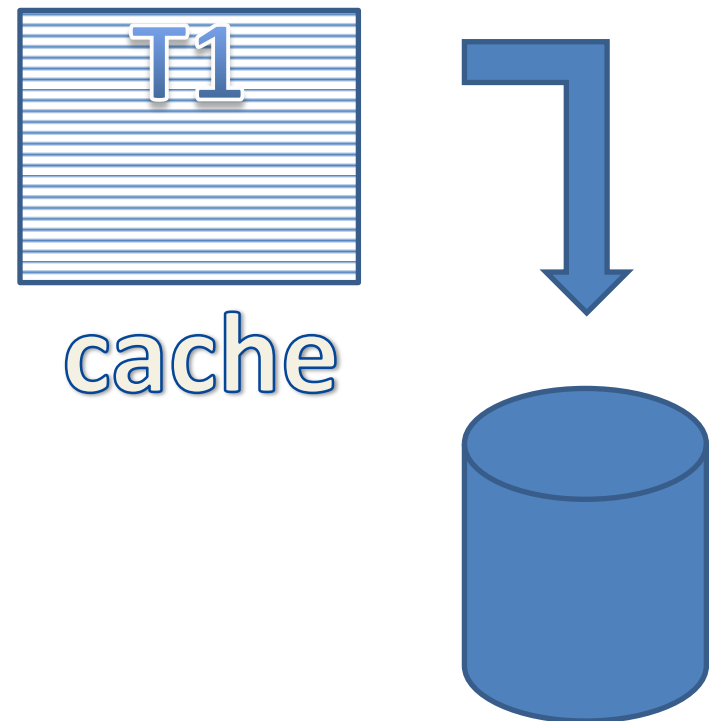
Sobre logs

- Registro adiantado em Log

Roubado/Não-Roubado

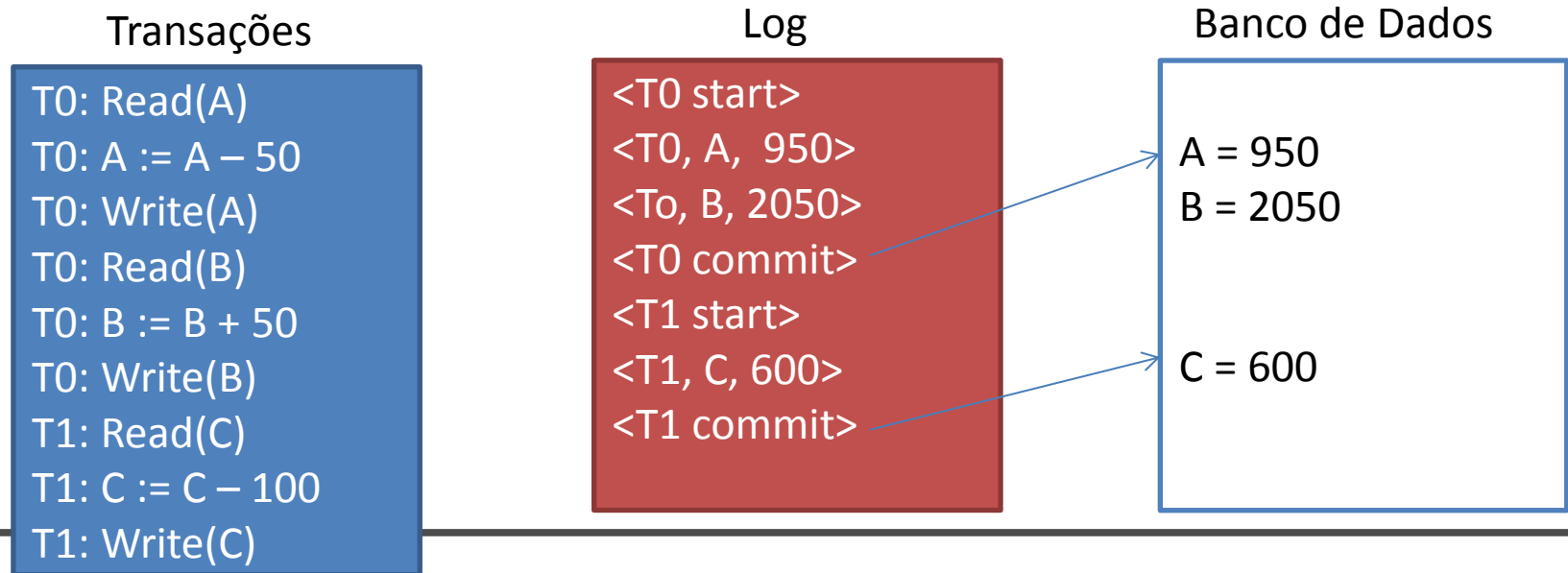


Forçado/Não-Forçado



Modificações adiadas ao Banco de Dados

- Garante atomicidade de transações quando todas as modificações do BD são escritas no Log
 - Adia a execução de todas as operações write de uma transação até sua efetivação parcial.



- Define o valor de todos os itens de dados atualizados pela transação Ti para os novos valores.
- A operação é **idempotente**, isto é, executá-la várias vezes geram o mesmo resultado.
- Após a ocorrência de uma falha, o subsistema de recuperação consulta o log para determinar quais transações têm de ser refeitas.
 - Ti será refeita, se e somente se, o log contiver os registros <Ti start> e <Ti commit>

Modificações imediatas de Banco de Dados

- Permite que as modificações no BD sejam enviadas enquanto as transações ainda estão no estado ativo: **Modificações não-efetivadas.**
- Na ocorrência de uma queda ou de uma falha de transação, o sistema deverá usar o campo relativo ao valor antigo dos registros do log para restauração dos itens de dados modificados.

Modificações imediatas de Banco de Dados

- Antes que uma transação T_i inicie sua execução, o registro $\langle T_i \text{ start} \rangle$ é escrito no log. Durante sua execução, qualquer operação de $\text{write}(X)$ feita por T_i é precedida pela escrita apropriada do novo registro no log. Quando T_i é parcialmente efetivada o registro $\langle T_i \text{ commit} \rangle$ é escrito no log.

Transações

```
T0: Read(A)
T0: A := A - 50
T0: Write(A)
T0: Read(B)
T0: B := B + 50
T0: Write(B)
T1: Read(C)
T1: C := C - 100
T1: Write(C)
```

Log

```
<T0 start>
<T0, A, 1000, 950>
<T0, B, 2000, 2050>
<T0 commit>
<T1 start>
<T1, C, 700, 600>
<T1 commit>
```

Banco de Dados

```
A = 950
B = 2050

C = 600
```

Procedimentos: Undo(Ti) e Redo(Ti)

- **Undo(Ti)** retorna aos valores antigos todos os itens de dados atualizados pela transação Ti
 - Se o log contém o registro <Ti start>, mas não tem o registro <Ti commit>
- **Redo(Ti)** ajusta os valores de todos os itens de dados atualizados pela transação para os novos valores.
 - Se o log contém tanto o registro <Ti start> quanto o registro <Ti commit>
- Operações são ***idempotentes***.

Undo/Redo

Questão 9. CESPE - 2012 - MPE-PI - Analista Ministerial - Informática - Cargo 6

A respeito de administração de banco de dados relacionais; administração de usuários e perfis de acesso; controle de proteção, integridade e concorrência; restauração de dados; tolerância a falhas e continuidade de operação e otimização de desempenho, julgue os itens de 85 a 87.

[85] A garantia de consistência do banco de dados, na ocorrência de uma queda de sistema enquanto uma transação T, para transferência de 50 dólares de uma conta A para uma conta B, estiver sendo executada, após o débito em A e antes do crédito em B, que resulte em perda dos conteúdos de memória, requer a manutenção das informações de modificações em um sistema de armazenamento estável.

Simplificando: Possíveis Estratégias

1. Perda por falha:

- Reconstrução (**REDO**)
- Backup -----> estado consistente mais próximo da falha

2. O BD tornou-se inconsistente:

- Reverter mudanças (**UNDO**)
- BD inconsistente -----> BD consistente

Checkpoints

- Diminui o overhead de leitura do log após uma falha introduzindo no log os pontos de controle.
- Exigem 3 ações
 1. Saída, para armazenamento estável, de todos os registros residentes na memória principal
 2. Saída, para disco, de todos os blocos de buffer modificados.
 3. Saída para armazenamento estável, de um registro de log **<checkpoint>**

Fuzzy checkpoint

- O tempo necessário para forçar a gravação de todos os buffers de memória modificados pode atrasar o processamento da transação.
- Fuzzy checkpoint – o sistema poderá reassumir o processamento das transações depois que o registro [checkpoint] for escrito no log

Paginação Shadow

- Uma alternativa às técnicas de recuperação baseada em log.
- Exige (teoricamente) menos acessos ao disco
- Difícil de ser aplicada em transações concorrentes.
- Pressupõe-se que o BD seja composto por “n” páginas de tamanho fixo
- Uma tabela de páginas com “n” entradas é construída

Paginação Shadow

- A ideia é manter 2 tabelas de página durante o processamento
 - A tabela de páginas *atuais*
 - A tabela de páginas *shadow*
- Quando um transação começa ambas as tabelas são idênticas.

Begin_Transaction:

Operações de Write(X) → Tabela de página corrente

Tabela de página corrente → Tabela de página imagem

(Nunca é modificada durante a transação)

Páginas do BD

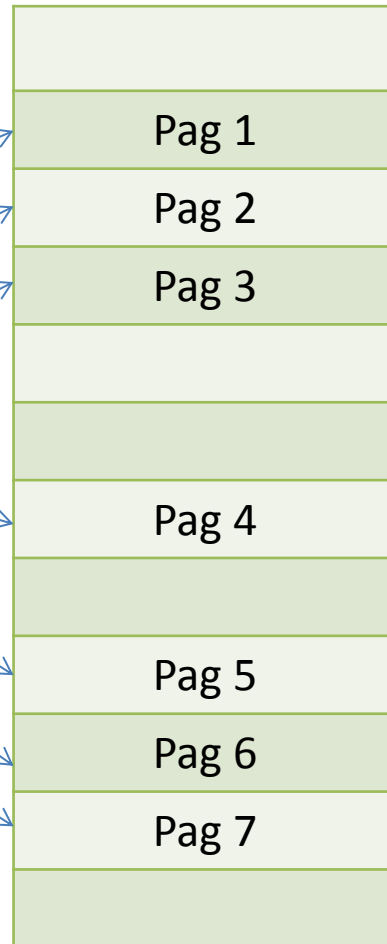
Páginas do BD

Tab_pag_corrente
(Iguar a imagem)

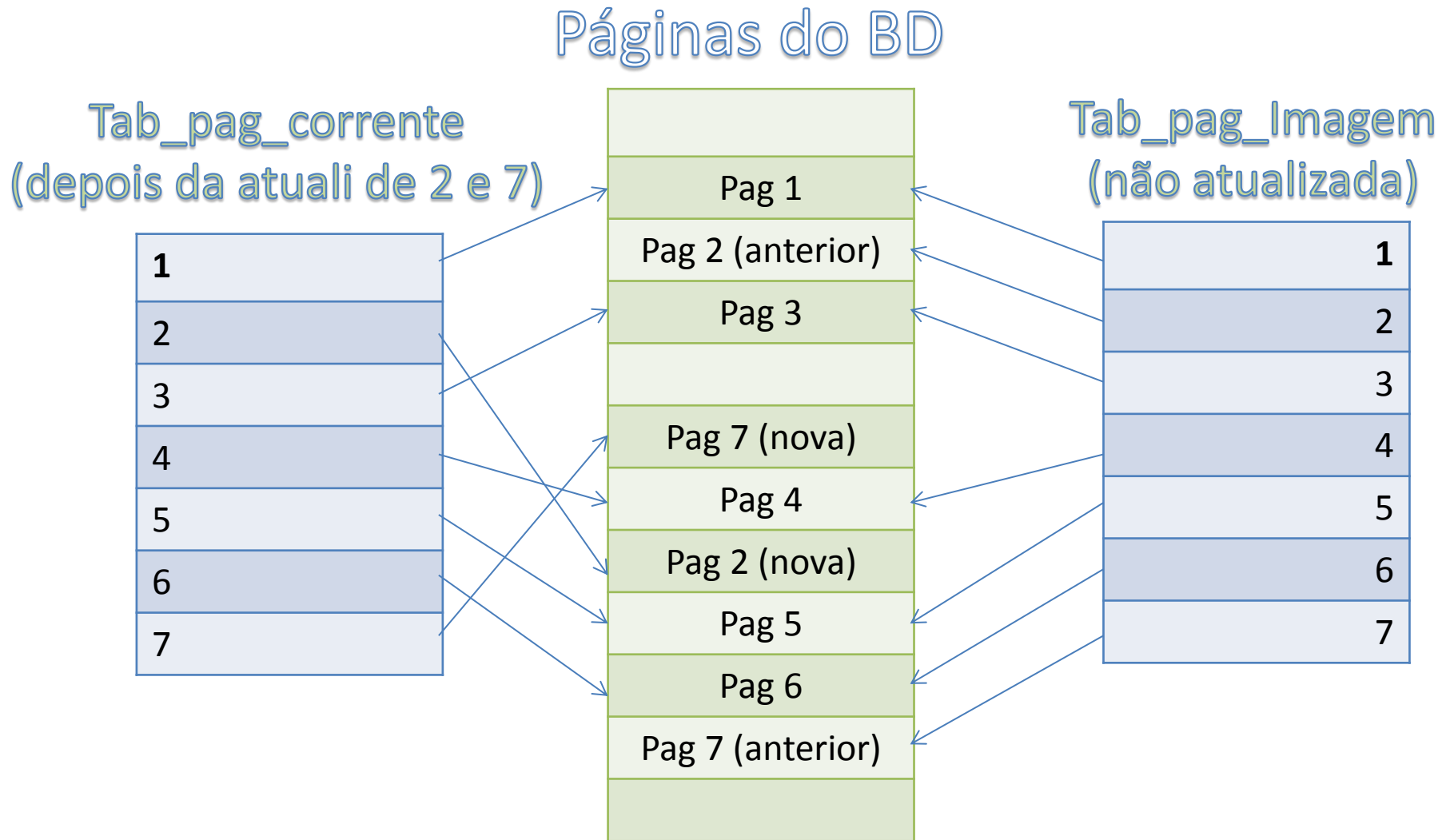
1
2
3
4
5
6
7

Tab_pag_Imagem
(não é modificada)

1
2
3
4
5
6
7



Páginas do BD



Páginas do BD

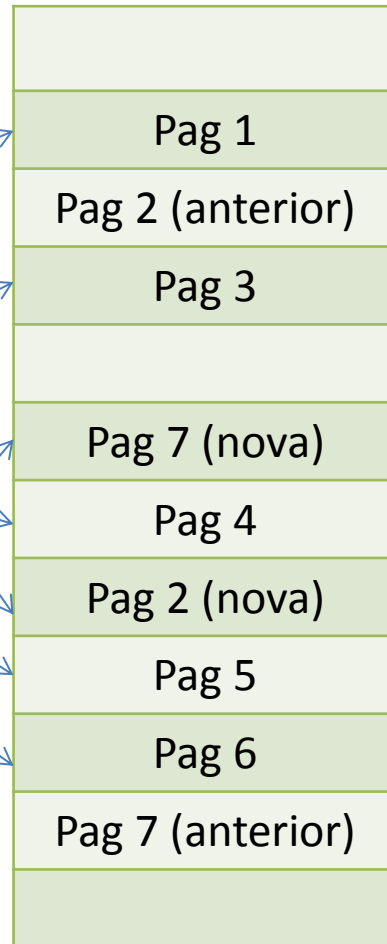
Páginas do BD

Tab_pag_imagem
(transação efetivada)

1
2
3
4
5
6
7

Tab_pag_corrente
(atualizada)

1
2
3
4
5
6
7



Páginas Imagem (Shadow)

- Ambiente Mono-usuário:
 - não é necessário o log
- Ambiente Multi-usuário:
 - pode-se usar o log se o método de controle de concorrência usar

Páginas Imagem (Shadow)

- Vantagem
 - Recuperação após falhas é significativamente mais rápida.
 - Não há necessidade de Undo ou Redo de operações
- Desvantagens
 - **Fragmentação de dado:** Páginas atualizadas mudam de localização no disco, impedindo de manter juntas páginas relacionadas
 - **Overhead de efetivação:** Se a tabela de páginas é grande, o tempo para gravar as tabelas de páginas imagem no commit é significativo
 - **Coleta de lixo:** Garbage Collection (liberação de páginas antigas) é necessário após o commit

Questão 10. CESPE - 2012 - MPE-PI - Analista Ministerial - Informática - Cargo 6

[86] Na paginação shadow, o banco de dados é particionado em um número de blocos de comprimento variável. Durante o processamento dessa transação, mantêm-se duas tabelas de páginas ditas atual e shadow. Se a transação for completada, a página atual é atualizada com a cópia da página shadow mantida em armazenamento volátil.

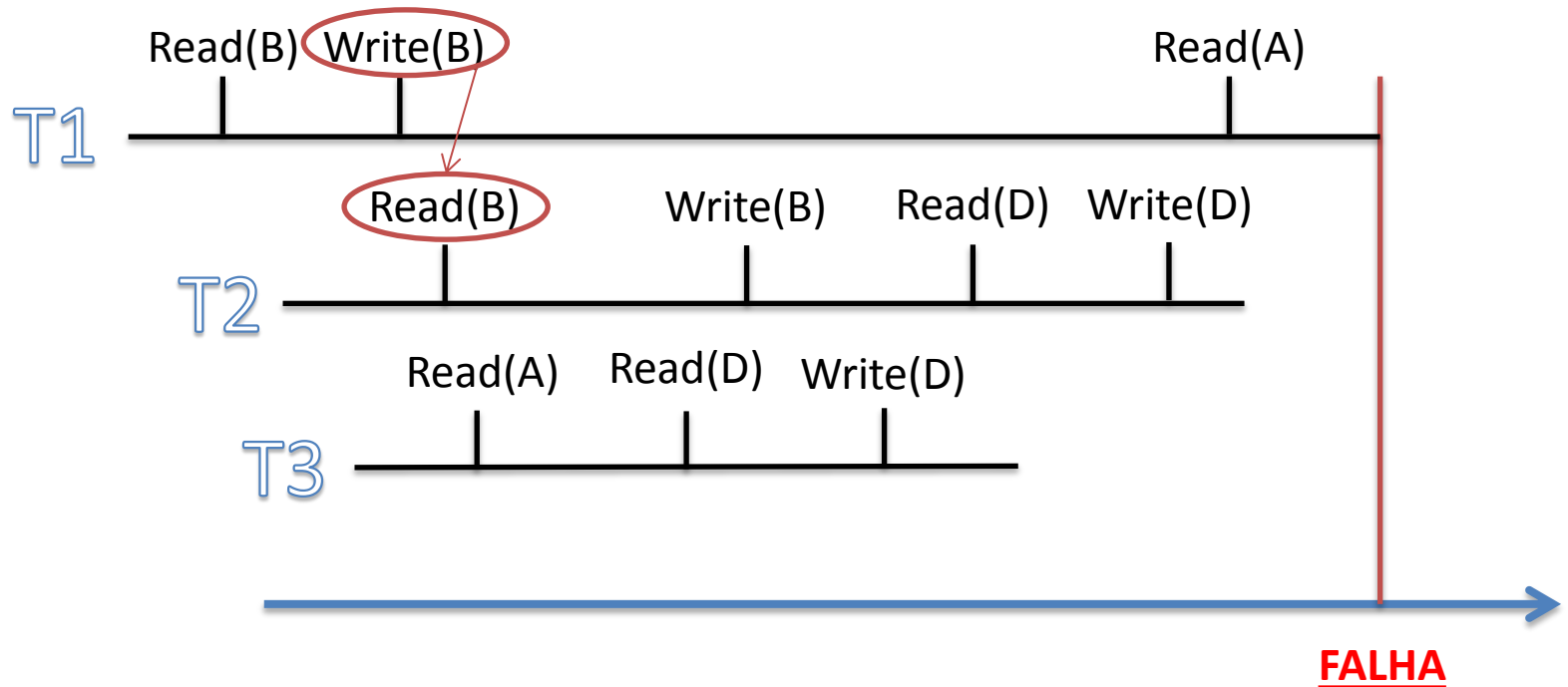
Estados de transações

- O SGBD deve manter uma lista de transações processadas com os seguintes estados
 - **Ativas:** iniciadas mas sem commit
 - **Acabadas (committed):** desde o último checkpoint
 - **Abortadas** desde o último checkpoint

Rollback

- **Rollback de Transações (Undo):** É necessário se uma falha ocorrer depois do BD ter sido atualizado
 - Qualquer item de dado atualizado pela transação deve voltar a seu valor anterior
- **Rollback em Cascata:** Se uma transação T é desfeita e uma transação S leu algum dado atualizado por T, S também tem que ser desfeita e assim por diante.
 - Podemos resolver esse problema exigindo o bloqueio de duas fases severo*

Rollback em Cascata ou UNDO



T1 é desfeita porque não alcançou o *commit*

T2 é desfeita porque leu o valor de B gravado por T1

T3 é refeita

Questão 11. CESPE - 2009 - TCU - Analista de Controle Externo - Tecnologia da Informação - Prova 2

T1	T2
lock(p1)	
read(p1)	
write(p1)	lock(p1)
lock(p2)	
unlock(p1)	
	read(p1)
	write(p1)
	lock(p2)
	unlock(p1)
read(p2)	
write(p2)	
unlock(p2)	
	read(p2)
	write(p2)
	unlock(p2)

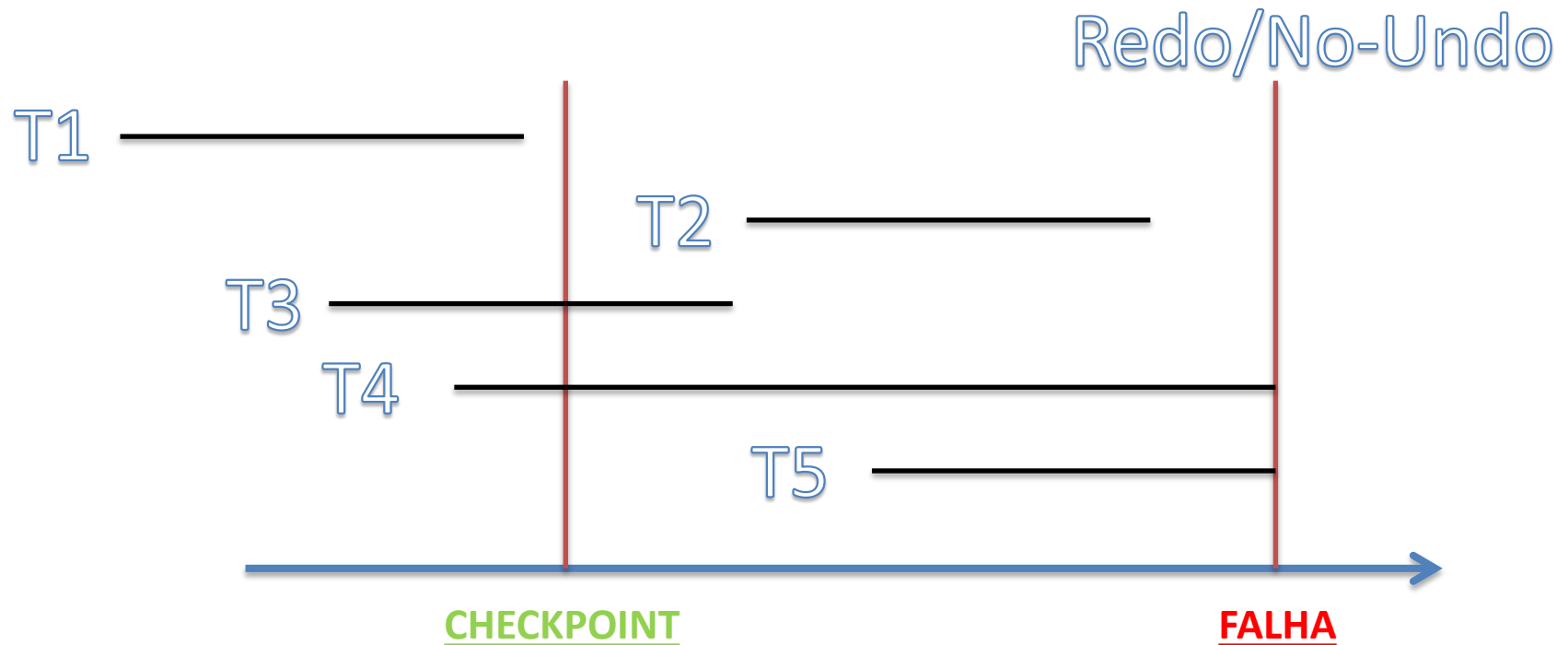
Considerando as transações T1 e T2 apresentadas acima, julgue o item subsequentes.

[130] Se, por alguma razão, a transação T1 for abortada, o requerimento de atomicidade implicará que também sejam abortadas as transações como T2, aquelas que igualmente tenham visto os resultados de T1, transações que tenham visto os writes dessas transações e assim por diante. Esse problema é conhecido como rollback em cascata

Recuperação com Atualização Adiada (RAA)

- Ambiente Multiusuário
 - Protocolo
 - Depende do protocolo usado no controle de concorrência
 - Assumir protocolo de duas fases com bloqueios antes do início da execução, guardando-os até o commit
 - Procedimento
 - Usar duas listas de transações: as transações acabadas desde o último checkpoint e as transações ativas
 - Aplicar a operação Redo para todas as operações write_item das transações acabadas no log, na ordem na qual elas foram gravadas
 - As transações ativas e não acabadas são canceladas e devem ser re-submetidas

Recuperação com Atualização Adiada (RAA)



T1: OK

T2 e T3: operações write_item devem ser refeitas

T4 e T5: canceladas e re-submetidas

Recuperação com Atualização Adiada (RAA)

- Desvantagem
 - Limita a execução concorrente das transações porque itens ficam bloqueados até o commit das transações
- Vantagens
 - Uma transação não grava as modificações no BD até o commit. Logo, uma transação nunca é desfeita por causa de falha
 - Uma transação nunca vai ler o valor de um item gravado por outra não acabada, porque os itens estão bloqueados. Logo, não ocorrerá rollback em cascata

Recuperação com Atualização Imediata (RAI)

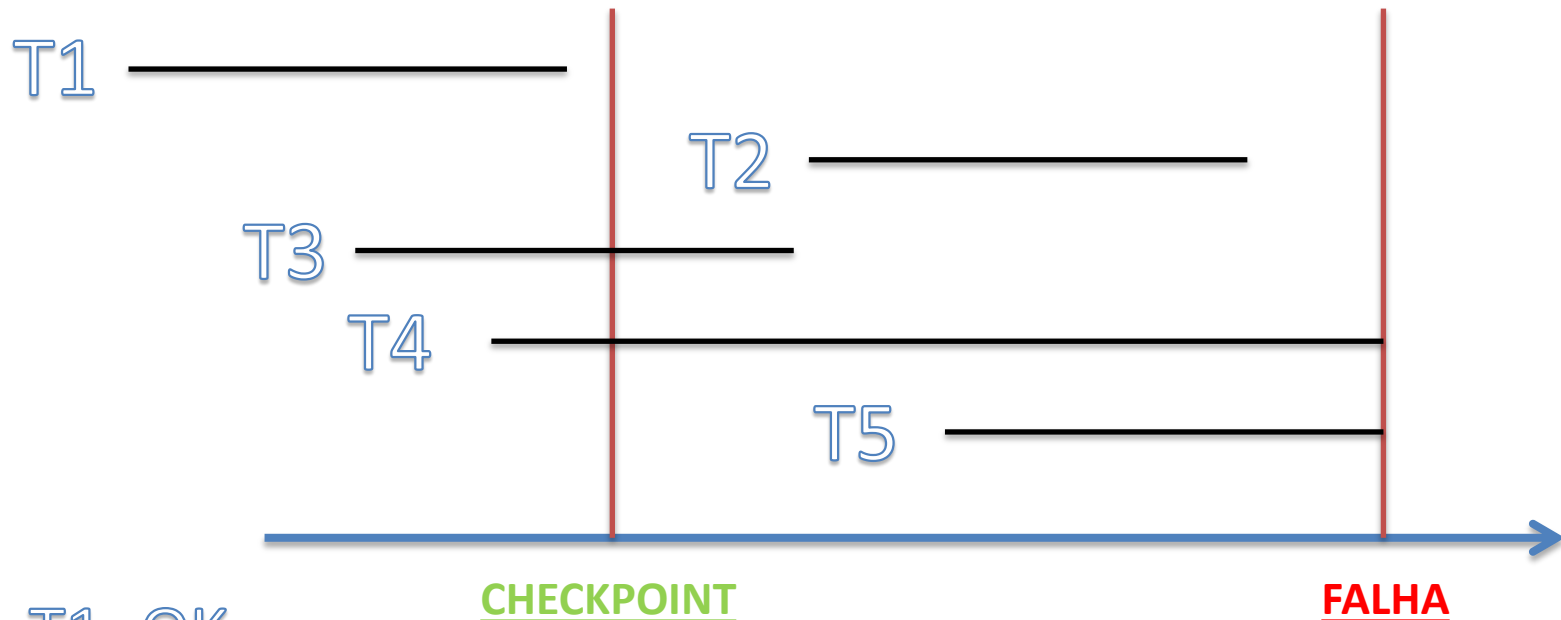
- Dois tipos de Algoritmos
 - Undo/No-Redo
 - Se a técnica de recuperação garante que todas as atualizações são gravadas no BD (disco) antes do commit da transação, não é necessário Redo
 - Undo/Redo
 - Se as modificações são gravada no BD (disco) depois do commit da transação
 - Caso mais geral e mais complexo

Recuperação com Atualização Imediata (RAI)

- Ambiente Multiusuário
 - Protocolo
 - Assume-se que o log inclui checkpoints e o protocolo de concorrência como o de duas fases
 - Procedimento
 - Usar duas listas de transações: as transações acabadas desde o último checkpoint e as transações ativas
 - Aplicar a operação Undo para todas as operações write_item das transações ativas, na ordem inversa de suas gravações no log
 - Aplicar a operação Redo para todas as operações write_item das transações acabadas, na ordem na qual elas foram gravadas no log

Recuperação com Atualização Imediata(RAI)

Redo/Undo



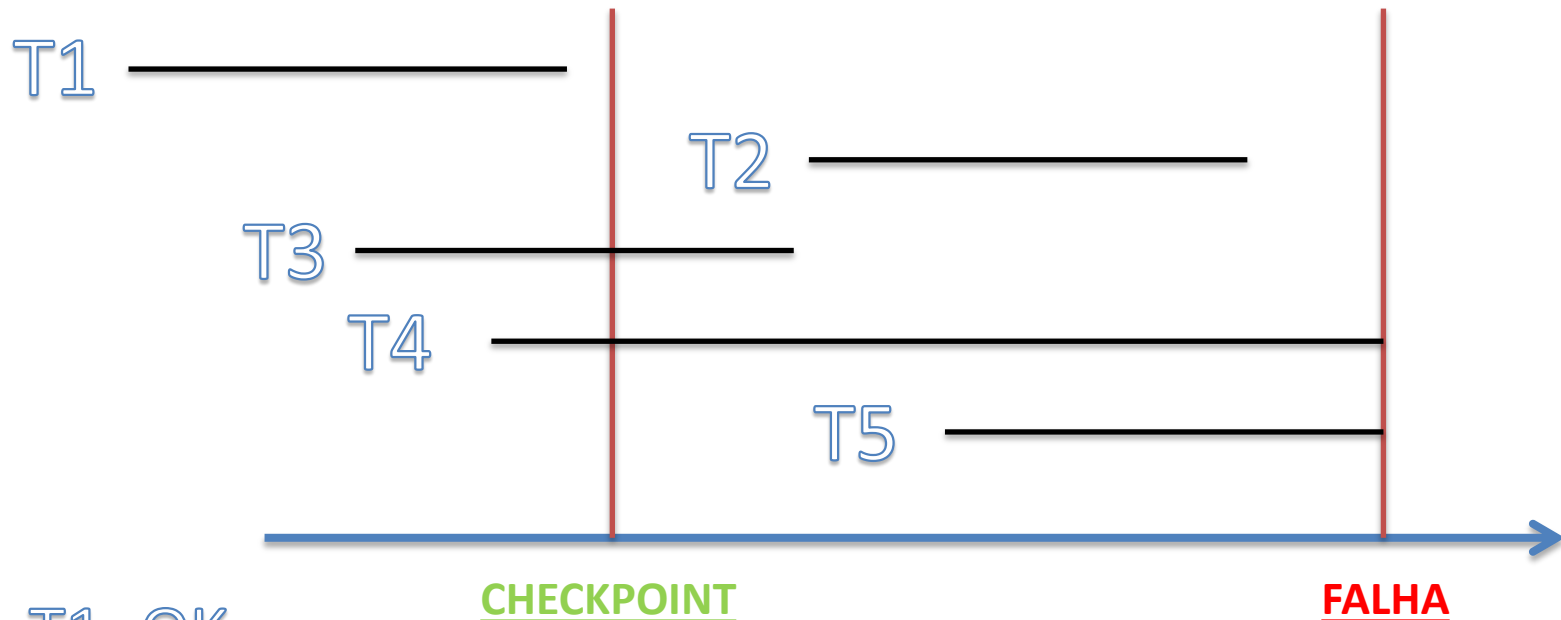
T1: OK

T2 e T3: operações write_item devem ser refeitas

T4 e T5: desfeitas e re-submetidas

Recuperação com Atualização Imediata(RAI)

No-Redo/Undo



T1: OK

T2 e T3: operações write_item não devem ser refeitas

T4 e T5: desfeitas e re-submetidas

VISÃO GERAL DO ARIES

- O ARIES é um algoritmo de recuperação que é projetado para trabalhar com uma abordagem de “roubar” e “não forçar”.
- Está baseado em três conceitos:
 1. Registro adiantado em log
 2. Repetição de histórico durante o refazer
 3. Mudanças do log durante o desfazer



Fases do algoritmos de Aries

- Quando o gerenciador de recuperação é invocado após uma falha, o reinício se procede em três fases:
 - **Fase de Análise(1)**: identifica páginas sujas no buffer e transações ativas no momento da falha.
 - **Fase de Refazer(2)**: repete todas as ações, começando do ponto apropriado no log e restaura o estado da base de dados idêntico ao momento da falha.
 - **Fase de Desfazer(3)**: desfaz as ações das transações que não realizaram o commit, de forma que a base de dados reflita apenas as ações das transações que realizaram o commit.

Mais informações sobre o ARIES

- Informações necessárias
 - Log, Tabela de Transações e Tabela de Pagina Lixo.
 - Essas duas tabelas são mantidas pelo gerenciador de transações e gravadas no log durante o checkpoint.
- Cada registro tem um numero de sequência de log (LSN)
- Usa fuzzy checkpoint



Exemplo do ARIES

(a)

Lsn	Ultimo_lsn	Id_transacao	Tipo	Id_pagina	Outra_informacao
1	0	T_1	update	C	...
2	0	T_2	update	B	...
3	1	T_1	commit		...
4	begin check point				
5	end check point				
6	0	T_3	update	A	...
7	2	T_2	update	C	...
8	7	T_2	commit		...

(b)

TABELA DE TRANSAÇÕES

Id_transacoes	Ultimo_lsn	Status
T_1	3	commit
T_2	2	in progress

TABELA DE PÁGINAS SUJAS

Id_pagina	Lsn
C	1
B	2

(c)

TABELA DE TRANSAÇÕES

Id_transacoes	Ultimo_lsn	Status
T_1	3	commit
T_2	8	commit
T_3	6	in progress

TABELA DE PÁGINAS SUJAS

Id_pagina	Lsn
C	1
B	2
A	6



Questão 12. FCC - 2010 - TCE-SP - Agente da Fiscalização Financeira - Informática - Produção e BD

[78]. Tratando-se do algoritmo ARIES, quando o gerenciador de recuperação é invocado após uma falha, o reinício se procede em fases. A identificação das páginas sujas no buffer pool e das transações ativas no momento da falha pertence à fase de

- (A) preempção.
- (B) refazer.
- (C) lock.
- (D) análise.
- (E) correção.

Backup e Recuperação após Falhas Catastróficas

- Exemplo: pane no disco
 - Principal Técnica: backup do BD
 - Cópia periódica do BD e do log em outro meio de armazenamento (fitas)
 - É necessário backup do log para não perder as transações efetuadas desde o último checkpoint
 - Um log é inicializado após cada operação de backup. Logo, para recuperar após uma falha no disco
 - O BD é recriado a partir do último backup
 - Os efeitos de todas as transações committed, cujas operações foram gravadas no log, são reconstruídos

Questões pro bancas



CESPEUnB

UNIVERSIDADE DE BRASÍLIA

Questão 13. CESPE 2012 – TRT - 10ª Região – 2012 – Analista de Sistemas

Com relação a sistemas de gerenciamento de banco de dados, julgue os itens seguintes.

[100] Os algoritmos undo-redo, buferização e fuzzy de checkpoint consistem em técnicas de realização de backup de bancos de dados.

[101] A consistência, uma propriedade que deve ser garantida em um sistema de gerenciamento de banco de dados, ocorre em situações em que todas as ações de uma unidade de trabalho são efetivadas (commit) sem que haja falha na unidade de trabalho. Caso haja falha, é realizado rollback.

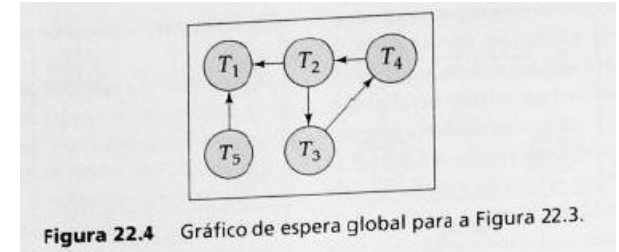
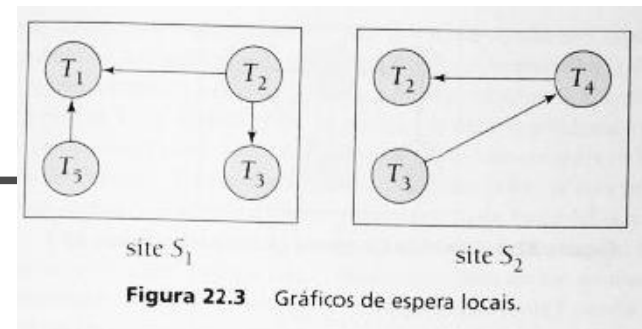
[102] O protocolo de bloqueio é utilizado para resolver problemas relacionados à concorrência.

Questão 14. CESPE - 2012 - MPE-PI - Analista Ministerial - Informática - Cargo 6

[87] Em sistemas distribuídos que utilizam técnicas de prevenção de impasses, o fato de um gráfico de espera ser acíclico indica a ausência de impasse, mesmo que a união dos gráficos locais gerem um gráfico global cíclico

Tratamento de impasses

- Gráficos locais
- Gráfico de espera global
- Dois esquemas mais comuns
 - Versão centralizada (coordenador da detecção de impasses.
 - Versão totalmente distribuída (todos os coordenadores assumem a responsabilidade pela detecção do impasse) → representando parte do gráfico total.



Questão 15. CESPE - 2010 - Banco da Amazônia - Técnico Científico – Tec. da Inf. – Arq. de Tecnologia

A respeito de banco de dados, julgue os itens que se seguem.

[89] No controle de transações, é desnecessário que a atomicidade de cada transação tome conhecimento de outras transações concorrentes no sistema.

[101] Consistência é considerada uma propriedade das transações para os sistemas de bancos de dados.

Há dois motivos para permitir a concorrência de transações:

1. Uma transação consiste em diversos passos que podem envolver atividades de I/O ou CPU.
 - Estes podem operar em paralelo e este paralelismo explorado para executar diversas transações em paralelo.
2. Transações curtas e longas podem ser executadas em paralelo se estiverem operando em diferentes partes do banco.
 - Imagina se tivesse que esperar a transação longa para poder executar a curta?

Questão 16. CESPE - 2008 - STJ - Analista Judiciário - Tecnologia da Informação

Acerca de arquiteturas de aplicações cliente-servidor e para a Internet, julgue os próximos itens.

[86] Para recuperar-se de falhas que afetam transações, um SGBD pode manter um log, de modo a controlar as operações da transação que afetem valores dos itens do banco de dados.

Quando de uma falha, o SGBD deve buscar, no log, as transações T com registro [start_transaction,T], mas que não tenham o registro [commit,T]. Essas transações serão revertidas (rollback), de modo a possibilitar que seus efeitos sejam desfeitos no banco de dados.

Questão 17. CESPE - 2009 - TCU - Analista de Controle Externo - Tecnologia da Informação - Prova 2

A respeito de sistemas transacionais, julgue os itens a seguir.

[126] Um escalonamento é considerado correto quando se pode encontrar um escalonamento serial que seja equivalente a ele. Dado um conjunto de transações T_1, \dots, T_n , dois escalonamentos S_1 e S_2 são equivalentes se existe sincronização read-write e sincronização write-write.

Escalonamento ou História de Transações

- A ordem de execução das operações de várias transações executando concorrentemente na forma intercalada
 - Escalonamento e Recuperabilidade
 - Um escalonamento S de ' n ' transações T_1, T_2, \dots, T_n é um ordenamento das operações dessas transações, sujeito à restrição de que para cada transação T_i em S , as operações de T_i em S devem aparecer na mesma ordem que ocorrem em T_i

Conceitos relacionados a escalonamento

- Operações Conflitantes em um escalonamento
 - Pertencem a diferentes transações, acessam o mesmo item e uma das duas operações é um **write_item**
- Escalonamento Recuperável
 - Nenhuma transação T em S é efetivada até que todas as transações T' que escrevam um item a ser lido por T , tenham sido efetivadas
- Rollback em Cascata
 - Uma transação não efetivada tem que ser rolled back porque lê um item de uma transação que falhou

Conceitos relacionados a escalonamento

- Escalonamento que Evita Rollback em Cascata
 - Toda transação no escalonamento apenas lê itens que foram escritos por transações efetivadas
- Escalonamento Estrito
 - Nenhuma transação pode ler ou escrever itens que a última transação que o escreveu não tenha ainda sido efetivada ou abortada

Serializabilidade de Escalonamentos

- Sem intercalação, se duas transações são submetidas, então
 - Todas as operações da transação T1 (em sequência) são seguidas por todas as operações de T2, ou vice-versa
- Teoria da Serializabilidade
 - Tenta determinar quais escalonamentos são “corretos” e quais não são e desenvolver técnicas que permitam apenas escalonamentos corretos

Escalonamentos

- Escalonamento Serial
 - As operações de cada transação são executadas consecutivamente, sem operações intercaladas com as de outras transações
- Escalonamento Não-serial
 - As operações de uma transação são executadas intercaladas com operações de outra transação
- **Escalonamento Serializável**
 - É equivalente a algum escalonamento serial com as mesmas 'n' transações

Sincronização de read-write e write-write

- Escalar operações conflitantes
 - que operam sobre o mesmo conjunto de dados:
read-write, write-write

BEGIN_TRANSACTION x = 0; x = x + 1; END_TRANSACTION	BEGIN_TRANSACTION x = 0; x = x + 2; END_TRANSACTION	BEGIN_TRANSACTION x = 0; x = x + 3; END_TRANSACTION
(a)	(b)	(c)

Schedule 1	x = 0; x = x + 1; x = 0; x = x + 2; x = 0; x = x + 3	Legal
Schedule 2	x = 0; x = 0; x = x + 1; x = x + 2; x = 0; x = x + 3;	Legal
Schedule 3	x = 0; x = 0; x = x + 1; x = 0; x = x + 2; x = x + 3;	Illegal

(d)

a) – c) Três transações T_1 , T_2 , e T_3
d) Possíveis escalonamentos

Questão 18. CESPE - 2009 - TCU - Analista de Controle Externo - Tecnologia da Informação - Prova 2

A respeito de sistemas transacionais, julgue os itens a seguir.

[127] No two-phase locking, os locks de dados supõem que a transação está dividida em uma fase de crescimento, na qual os locks são feitos, e em uma fase de encolhimento, na qual os locks são confirmados.

Questão 19. CESPE - 2009 - TCU - Analista de Controle Externo - Tecnologia da Informação

A respeito de sistemas transacionais, julgue os itens a seguir.

[128] As transações aninhadas devem ter a propriedade de ser serializáveis em relação aos seus pares, ou seja, o acesso a recursos compartilhados entre essas transações deve obedecer as regras de sincronização de read-write e write-write.

Questão 20. CESPE - 2008 - STF - Analista Judiciário - Tecnologia da Informação

O armazenamento e a recuperação de grandes quantidades de dados é um trabalho importante e muito explorado em um sistema gerenciador de banco de dados (SGBD). Com relação aos conceitos que envolvem esse sistema, julgue os itens que se seguem.

[92] A execução de transações de maneira concorrente possibilita o surgimento de inconsistências dos dados armazenados em um banco de dados. A responsabilidade pela consistência dos dados é única e exclusiva do banco de dados, mais especificamente, do componente de controle de concorrência

Questão 21. CESPE - 2012 - MPE-PI - Analista Ministerial - Informática - Cargo 6

A respeito de integridade e concorrência em banco de dados, julgue os itens subsecutivos.

[67] Devido ao fato de a concorrência de acessos a um banco de dados diminuir conforme aumenta a importância do controle da concorrência, os registros de acessos em memória, com o intuito de manter a integridade dos dados, geram um despejo na fila de acesso.

[68] Em banco de dados relacionais, o controle de concorrência é ilimitado para leituras de dados. Entretanto, modificações simultâneas disparadas ao mesmo dado por diferentes usuários são garantidas, pois, sempre que um dado estiver em modo protegido, o acesso restrito ao dado somente será permitido após autenticação



Questão 22. CESPE - 2011 - Correios - Analista de Correios - Analista de Sistemas – Produção

Com relação a banco de dados, julgue os itens a seguir

[88] Em um sistema gerenciador de banco de dados, o controle de concorrência consegue monitorar as modificações simultâneas de forma física e lógica, podendo a forma lógica ser feita por meio de linguagem de consulta

Questão 23. CESPE - 2010 - Banco da Amazônia - Técnico Científico – Tec. da Informação - BD

Acerca dos mecanismos de segurança e controle de concorrência dos sistemas de banco de dados, julgue os itens que se seguem.

[85] Atomicidade, consistência, isolamento e durabilidade são propriedades fundamentais que devem ser apresentadas por uma transação de banco de dados.

[86] O isolamento de uma transação é uma propriedade que garante que a transação seja encarada como uma unidade atômica de trabalho: ou todas as suas modificações de dados são executadas ou nenhuma delas é executada.

Questão 24. CESPE - 2010 - Banco da Amazônia - Técnico Científico – Tec. da Informação - BD

Acerca dos mecanismos de segurança e controle de concorrência dos sistemas de banco de dados, julgue os itens que se seguem.

[87] Um update lock insere um bloqueio no objeto para impedir que outros usuários façam alterações, porém não é ativado quando já existe um shared lock ou um exclusive lock no mesmo objeto.

[88] Uma transação do tipo dirty read lê todos os registros, não importando se estão sendo modificados ou se ainda não houve um commit.

Questão 25. CESPE - 2006 - ANATEL - Especialista em Regulação – Informática

A respeito do desenvolvimento de aplicações e bancos de dados, julgue os seguintes itens.

[94] No SQL, são possíveis os seguintes níveis de isolamento entre transações: serializável, read com efetivação, read sem efetivação e read repetitivo. No read com efetivação, não é possível a leitura de dados não-efetivados, nem é possível que, entre duas leituras de um registro feitas por uma transação, o registro tenha sido modificado por transações efetivadas com sucesso.

Níveis de isolamento

Isolation level	Dirty reads	Non-repeatable reads	Phantoms
Read Uncommitted	may occur	may occur	may occur
Read Committed	-	may occur	may occur
Repeatable Read	-	-	may occur
Serializable	-	-	-

Isolation level	Write Lock	Read Lock	Range Lock
Read Uncommitted	V	-	-
Read Committed	V	S	-
Repeatable Read	V	V	-
Serializable	V	V	V

"V" (also called "**exclusive lock**")

"S" Shared locks

Fundação Carlos Chagas

Questão 26. FCC - 2007 - MPU - Analista de Informática - Banco de Dados

[50] Um mecanismo de controle de concorrência utilizado pelos sistemas gerenciadores de bancos de dados é denominado

- (A) dependência sem commit.
- (B) deadlock.
- (C) atualização perdida.
- (D) bloqueio.
- (E) análise inconsistente.

Questão 27. FCC - 2010 - TCE-SP - Agente da Fiscalização Financeira - Informática - Produção e BD

[79]. As fases de operação do algoritmo ARIES são em número de

- (A) 1.
- (B) 2.
- (C) 3.
- (D) 4.
- (E) 5.

Questão 28. FCC - 2010 - TCE-SP - Agente da Fiscalização Financeira - Informática - Produção e BD

80. O sistema ARIES é considerado um adequado mecanismo de recuperação de BD, pois

- I. fornece alto grau de paralelismo.
- II. usa a abordagem não-tomar-memória e forçar escrita.
- III. reduz a sobrecarga na operação de registro no histórico-de-ocorrências.
- IV. minimiza o tempo de recuperação.

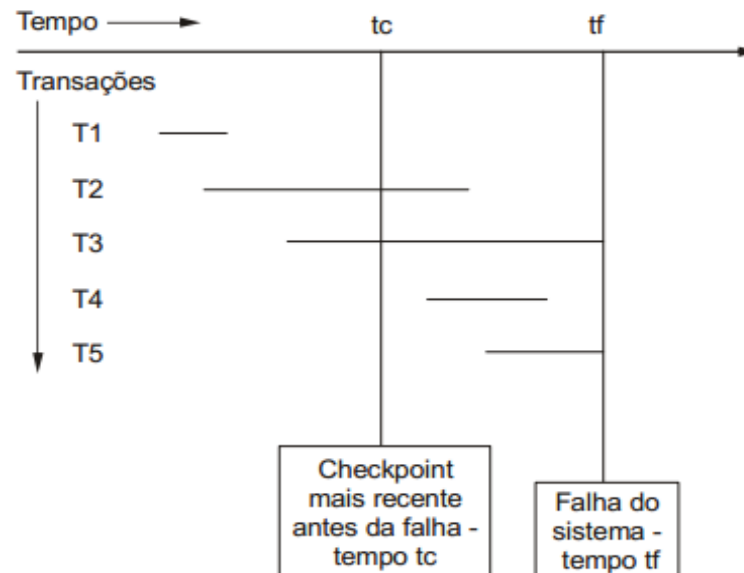
Está correto o que consta em

- (A) I e II, apenas.
- (B) I e III, apenas.
- (C) I, III e IV, apenas.
- (D) II, III e IV, apenas.
- (E) I, II, III e IV.

Questão 29. FCC - 2010 - TCE-SP - Agente da Fiscalização Financeira - Informática - Produção e BD

Instruções: Para responder às questões de números 76(30) e 77(31), considere os dados abaixo.

T1 a T5 são tipos de transações nas situações de execução (início e término), na linha do tempo, conforme demonstrado no gráfico.



Questão 29. FCC - 2010 - TCE-SP - Agente da Fiscalização Financeira - Informática - Produção e BD

[76]. Quando o sistema é reinicializado após a falha, as transações que devem ser desfeitas (D) e as que deve ser refeitas (R) são:

- (A) T1 (D), T2 (D), T3 (D), T4 (R) e T5 (R).
- (B) T3 (D), T4 (D), T1 (R) e T2 (R).
- (C) T3 (D), T5 (D), T2 (R) e T4 (R).
- (D) T3 (D), T4 (D) e T5 (R).
- (E) T4 (D), T1 (R), T2 (R) e T3 (R).

Questão 30. FCC - 2010 - TCE-SP - Agente da Fiscalização Financeira - Informática - Produção e BD

[77]. No momento da reinicialização, o sistema deve passar primeiro por alguns procedimentos, a fim de identificar todas as transações APENAS do tipo

- (A) T1 a T2.
- (B) T1 a T3.
- (C) T2 a T4.
- (D) T2 a T5.
- (E) T3 a T5.



Questão 31. FGV - 2008 - Senado Federal - Analista de Sistemas

[50] No que diz respeito às técnicas de controle de concorrência em bancos de dados, é correto afirmar que:

- (A) sempre que duas ou mais transações simultâneas tentam acessar o mesmo item de dados, elas provocam um deadlock no sistema de gerenciamento de banco de dados.
- (B) uma das técnicas para se detectar um estado de deadlock é construir um grafo wait-for.
- (C) o problema de espera indefinida associado a uma transação (starvation) ocorre sempre que o seu esquema de espera para itens bloqueados atribui prioridade maior aos requisitos de outras transações simultâneas cujas execuções provoquem repetidos aborts na execução de suas atualizações.
- (D) as técnicas de controle de concorrência baseadas em ordenação por timestamp não usam bloqueios e, por isso, diminuem as chances de ocorrência de deadlocks.
- (E) diz-se que uma transação segue o protocolo de bloqueio de duas fases se todas as operações (read_lock, write_lock) precedem a última operação de desbloqueio na transação

Questão 32. FGV - 2008 - Senado Federal - Analista de Sistemas

[49] Considere as seguintes afirmativas a respeito das propriedades ACID (Atomicidade, Consistência, Isolamento, Durabilidade) de uma transação T de banco de dados:

- I. A atomicidade garante que todos os efeitos da transação T sejam integralmente refletidos no banco de dados ou que, em caso de falha, nenhum deles seja aceito.
- II. A consistência garante que, se o banco de dados estiver inicialmente em estado inconsistente, a execução da transação T, por si só, é capaz de deixar o banco de dados em um estado consistente.
- III. O isolamento garante que a execução da transação T seja isolada da execução de quaisquer outras transações simultâneas, de modo a evitar que o banco de dados venha a assumir um estado inconsistente.
- IV. A durabilidade garante que as atualizações promovidas pela transação T no estado do banco de dados sejam sempre mantidas, mesmo que ocorra uma falha no sistema durante a sua execução.

Questão 32. FGV - 2008 - Senado Federal - Analista de Sistemas

Assinale:

- (A) se apenas as afirmativas I, II e III estiverem corretas.
- (B) se apenas as afirmativas I e II estiverem corretas.
- (C) se apenas as afirmativas II, III e IV estiverem corretas.
- (D) se apenas as afirmativas I e III estiverem correta
- (E) se todas as afirmativas estiverem corretas



ESAF

Escola de Administração Fazendária

Questão 33. ESAF - 2005 - Receita Federal - Auditor Fiscal da RF - Área Tecnologia da Informação - Prova 3

[15]- Analise as seguintes afirmações relacionadas às capacidades que devem ser atendidas pelos bancos de dados orientados a objetos:

I. Concorrência: o SGBD orientado a objetos não necessita nem implementa controle de concorrência, que deve ser feita pelas transações solicitadas pelo aplicativo conectado ao banco.

II. Transações: unidades executadas inteiramente ou não. Se um conjunto de atualizações ocorreu dentro de uma transação, todas ou parte delas devem ser visíveis ao mundo exterior, independentemente do estado final da transação.

III. Recuperação: o SGBD deve garantir que os resultados parciais ou atualizações parciais de transações que falharam não sejam propagados para o banco de dados persistente.

Questão 33. ESAF - 2005 - Receita Federal - Auditor Fiscal da RF - Área Tecnologia da Informação - Prova 3

IV. Persistência: capacidade de objetos persistirem através de diferentes chamadas do programa. Dados manipulados por bancos de dados orientados a objetos devem ser persistentes ou transientes. Dados transientes são somente válidos dentro de um programa ou transação; eles são perdidos uma vez que o programa ou transação termine. Dados persistentes são armazenados fora de uma transação e sobrevivem às atualizações. Indique a opção que contenha todas as afirmações verdadeiras.

(A) I e II (B) II e IV (C) II e III (D) I e III (E) III e IV

Questão 34. ESAF - 2004 - CGU - Analista de Finanças e Controle - Área - Tecnologia da Informação

[21] Quando um sistema utiliza um Banco de Dados, o SGBD é responsável por controlar os acessos concorrentes aos dados compartilhados, prover meios para acesso aos dados e implementar mecanismos por meio dos quais seja possível manter a consistência dos dados em caso de falhas. Com relação ao controle do acesso a um Banco de Dados, é correto afirmar que os bloqueios

- a) exclusivos e compartilhados são usados para garantir que um recurso possa ser modificado.
- b) de atualização assinalam que uma transação pretende modificar um recurso. Várias transações podem manter simultaneamente um bloqueio de atualização em um mesmo recurso.
- c) exclusivos são usados para garantir que um recurso possa ser lido. Várias transações podem manter simultaneamente um bloqueio exclusivo em um mesmo recurso.

Questão 34. ESAF - 2004 - CGU - Analista de Finanças e Controle - Área - Tecnologia da Informação

[21] Quando um sistema utiliza um Banco de Dados, o SGBD é responsável por controlar os acessos concorrentes aos dados compartilhados, prover meios para acesso aos dados e implementar mecanismos por meio dos quais seja possível manter a consistência dos dados em caso de falhas. Com relação ao controle do acesso a um Banco de Dados, é correto afirmar que os bloqueios

- d) compartilhados são usados para garantir que um recurso possa ser lido. Nenhuma transação pode modificar os dados de um recurso enquanto existir um bloqueio compartilhado nesse recurso mantido por outra transação.
- e) exclusivos permitem que tanto o bloqueio compartilhado quanto o de atualização possam ser concedidos simultaneamente para o recurso em questão, desde que o de atualização aguarde a liberação do exclusivo para validar a modificação nos dados envolvidos.



Questão 35. CESGRANRIO - 2006 - Petrobrás - Analista de Sistemas Pleno - Infra-estrutura

[41] Considere as afirmativas abaixo sobre gerência de transações e indique a única correta.

- (A) A leitura suja ocorre quando uma transação A lê um dado modificado pela transação B, antes que B termine, e a transação B termina com sucesso.
- (B) Se o nível de isolamento do banco de dados for read committed, as alterações realizadas por uma transação em um dado serão imediatamente vistas por outras transações, mesmo antes do término da primeira transação.
- (C) Mesmo após a execução de um comando commit, é possível a uma transação retornar a seu estado inicial com a utilização de rollback.
- (D) Um escalonamento é a ordem de execução de várias transações de forma concorrente e possivelmente entrelaçada.
- (E) Em transações que ocorrem simultaneamente não existe qualquer garantia de que os dados lidos em um determinado momento são consistentes, independentemente do nível de isolamento utilizado.

Questão 36. CESGRANRIO - 2006 - Petrobrás - Analista de Sistemas Pleno - Infra-estrutura

[58] Para a gerência de bloqueios em bancos de dados são feitas as afirmativas a seguir.

I - Um modo de prevenir a ocorrência de deadlocks (impasses) é fazer com que todas as transações bloqueiem antecipadamente todos os itens de que precisam: se qualquer um dos itens não puder ser bloqueado, todos devem ser liberados. Embora previna a ocorrência de deadlocks, essa solução limita o nível de concorrência no sistema.

II - Bloqueios binários são aqueles que permitem dois estados (ou valores): locked (bloqueado) e unlocked (desbloqueado). Os bloqueios binários são sempre do tipo exclusivo: nenhuma transação diferente da que obteve o lock pode acessar os dados bloqueados.

III - Bloqueios podem ser obtidos em itens do banco de dados de diferentes granularidades, como, por exemplo: registros, blocos, tabelas e, até mesmo, todo o banco de dados. Tendo em vista que os itens de diferentes granularidades formam uma hierarquia em forma de árvore, uma mesma transação pode obter diferentes tipos de bloqueios nos nós, ao longo do caminho da raiz (banco de dados) até um nó específico da árvore.

Questão 36. CESGRANRIO - 2006 - Petrobrás - Analista de Sistemas Pleno - Infra-estrutura

IV - O protocolo de bloqueio em duas fases (two-phase locking) é composto pelas fases: (i) de expansão (ou crescimento), onde bloqueios são obtidos, mas nenhum bloqueio pode ser liberado, e (ii) de encolhimento (ou retrocesso), onde bloqueios existentes são liberados, mas nenhum novo bloqueio pode ser obtido.

V - Quando uma transação T1 obtém um bloqueio exclusivo sobre um item, nenhuma outra transação poderá alterar o dado bloqueado, sendo possível apenas ler tal dado. Quando uma transação T1 obtém um bloqueio do tipo compartilhado, será possível a outras transações ler e alterar o dado bloqueado, sendo somente proibida a exclusão de tal dado.

Estão corretas apenas as afirmativas:

(A) I e V (B) III e IV (C) I, III e IV (D) II, III e V (E) I, II, IV e V

Questão 37. CESGRANRIO - 2008 - CAPES - Analista de Sistemas

[69] No âmbito de banco de dados, considere os comportamentos:

I - Dirty Read;

II - Phantom Read;

III - Nonrepeatable Read.

No nível de isolamento REPEATABLE READ do padrão SQL-92, é(são) possível(eis) o(s) comportamento(s)

(A) I, apenas.

(B) II, apenas.

(C) III, apenas.

(D) I e II, apenas.

(E) I, II e III.

Questão 38. CESGRANRIO - 2008 - BNDES – Prof. Básico – Esp. – A. de Sistemas – Desenvolvimento

- [47] O Administrador de Dados de uma loja virtual verifica que há um problema no sistema controlador do estoque. Alguns produtos apresentam quantidades negativas, ou seja, houve mais vendas que a quantidade existente no estoque. O programador do sistema mostra, em pseudocódigo, o trecho que realiza a verificação do estoque.

```
BEGIN TRANSACTION

SELECT quantidade FROM estoque WHERE idproduto = <ID>;

IF quantidade > 0 THEN
    UPDATE estoque SET quantidade = quantidade - <QTD>
    WHERE idproduto = <ID>;
ELSE
    ERRO("quantidade insuficiente no estoque");
    ROLLBACK TRANSACTION;
END IF;

COMMIT TRANSACTION;
```

Questão 38. CESGRANRIO - 2008 - BNDES – Prof. Básico – Esp. – A. de Sistemas – Desenvolvimento

Qual opção descreve corretamente o problema e sua solução?

- (A) Está havendo um problema de leitura suja e a solução é mover o COMMIT TRANSACTION para dentro do THEN, logo abaixo do UPDATE.
- (B) Está havendo um problema de registro fantasma e a solução é alterar o nível de isolamento para serializável.
- (C) Está havendo um problema de deadlock e a solução é colocar a cláusula FOR UPDATE ao final do BEGIN TRANSACTION.
- (D) Está havendo um problema de concorrência e a solução é retirar a transação, eliminando os comandos BEGIN TRANSACTION, ROLLBACK TRANSACTION e COMMIT TRANSACTION.
- (E) A transação não bloqueia o produto consultado. O SELECT deverá ter, no final, a cláusula FOR UPDATE

Questão 39. CESGRANRIO - 2008 - BNDES - Profissional Básico – Esp. – A. de Sistemas – Des.

- [41] Duas transações (T1 e T2) de banco de dados executam as seguintes sequencias de operações:

T1:

Na tabela DEPARTAMENTO, bloqueia a linha N em modo compartilhado;

Na tabela DEPARTAMENTO, lê a coluna DESPESA da linha N;

Na tabela DEPARTAMENTO, desbloqueia a linha N;

Na tabela PROJETO, bloqueia a linha M em modo compartilhado;

Na tabela PROJETO, lê a coluna VERBA da linha M;

Na tabela PROJETO, desbloqueia a linha M;

Na tabela PROJETO, bloqueia a linha M em modo exclusivo;

Na tabela PROJETO, escreve a coluna VERBA da linha M com o valor VERBA + DESPESA;

Na tabela PROJETO, desbloqueia a linha M;

T2:

Na tabela PROJETO, bloqueia linha M em modo compartilhado;

Na tabela PROJETO, lê a coluna VERBA da linha M;

Na tabela PROJETO, desbloqueia a linha M;

Na tabela DEPARTAMENTO, bloqueia a linha N em modo compartilhado;

Na tabela DEPARTAMENTO, lê a coluna DESPESA da linha N;

Na tabela DEPARTAMENTO, desbloqueia a linha N;

Na tabela DEPARTAMENTO, bloqueia a linha N em modo exclusivo;

Na tabela DEPARTAMENTO, escreve a coluna DESPESA da linha N com o valor DESPESA + VERBA;

Na tabela DEPARTAMENTO, desbloqueia a linha N;

Questão 39. CESGRANRIO - 2008 - BNDES - Profissional Básico – Esp. - A. de Sistemas – Des.

É correto afirmar que essas transações

(A) não são serializáveis e, portanto, não podem ser executadas concorrentemente.

(B) não podem entrar em bloqueio infinito porque obedecem ao protocolo de bloqueio em duas fases (two-phase locking).

(C) podem entrar em bloqueio infinito (deadlock) se executadas concorrentemente.

(D) obedecem ao protocolo de bloqueio em duas fases (two-phase locking).

(E) são serializáveis e obedecem ao protocolo de bloqueio em duas fases (two-phase locking).

Questão 40. CESGRANRIO - 2006 - EPE - Técnico de Nível Superior - Área Tecnologia da Informação

[47] Para as técnicas de controle de concorrência em Bancos de Dados, assinale a afirmativa INCORRETA.

(A) O deadlock ocorre quando cada transação em um conjunto de duas ou mais transações espera por algum item que esteja bloqueado por alguma outra transação T no conjunto.

(B) Uma forma para detectar um estado de deadlock consiste em construir um grafo wait-for.

(C) O problema de espera indefinida (starvation) pode ocorrer se o esquema de espera para itens bloqueados for parcial, dando prioridade a algumas transações sobre as outras.

(D) As técnicas de controle de concorrência baseadas em ordenação por timestamp não usam bloqueios, portanto, deadlocks nunca podem ocorrer.

(E) Diz-se que uma transação segue o protocolo de bloqueio de duas fases se todas as operações (read_lock, write_lock) precedem a última operação de desbloqueio na transação.

Questão 41. CESGRANRIO - 2010 - EPE - Analista de Gestão Corporativa - Tecnologia da Informação

[41] Em determinada funcionalidade de um sistema interno de um órgão público, é necessário realizar uma transação em um banco de dados distribuído, segundo o protocolo Two-Phase Commit (2PC) ou o Three-Phase Commit (3PC), sobre os quais afirma-se que

- (A) ambos são bloqueantes.
- (B) existem dois nós coordenadores no 2PC.
- (C) na fase de expansão, ambos adquirem write-lock.
- (D) na fase de retração, o 3PC adquire write-lock.
- (E) somente o 2PC é bloqueante.

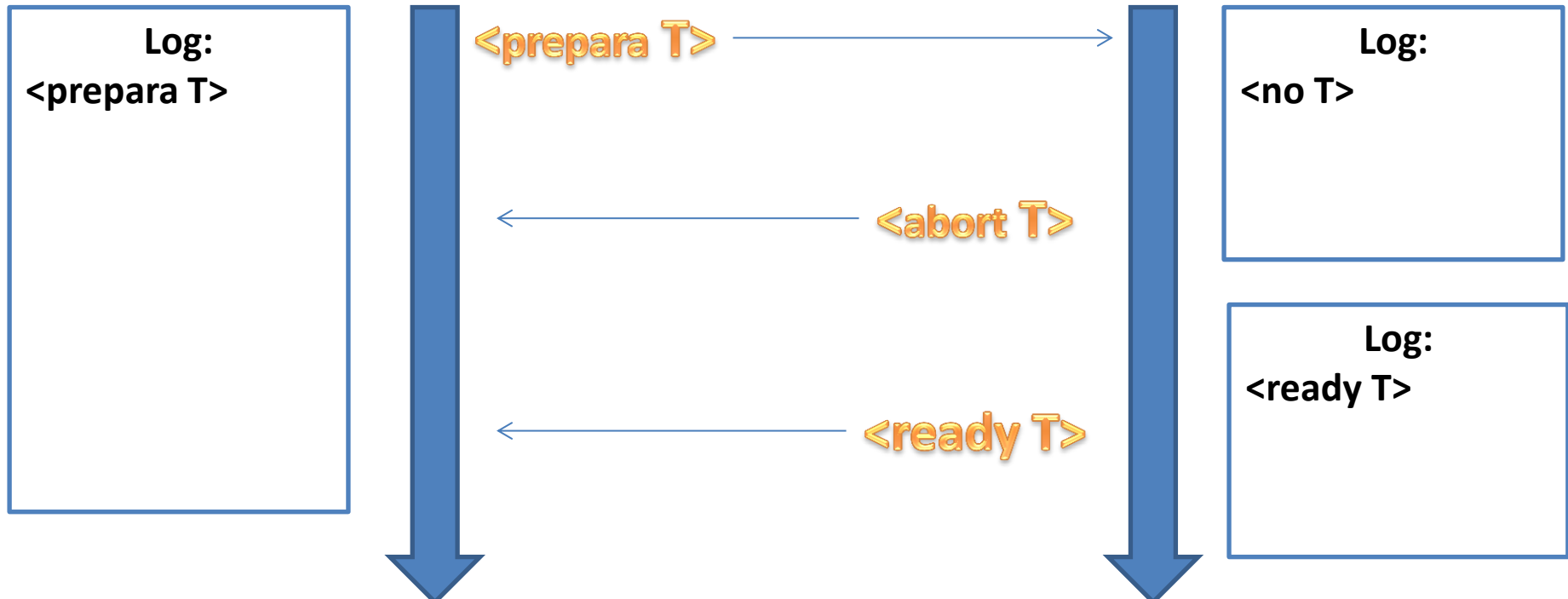
Protocolo de Efetivação em BDD

- Se desejarmos garantir a atomicidade, todos os sites envolvidos na execução de uma transação T devem concordar com o seu término:
 - Protocolo de efetivação em duas fases
 - Protocolo de efetivação em três fases
- A partir de agora quando falarmos de S1 e C1 estaremos nos referindo a Site e Coordenador.

Commit Protocol em duas fases

- Fase 01
 - C1 - Acrescenta o registro <**prepare** T> no log e força a manutenção do log em memória estável
 - C1 - Envia a mensagem **prepare** T para todos os sites em que T é executada.
 - S1 – Ao receber a mensagem, o gerenciador de transações do site determina se é possível efetivar sua porção da transação T
 - S1 – Se a resposta for negativa, ele adicionará um registro <no T> no log e enviará uma mensagem de **abort** T para C1
 - S1 – Se a resposta for positiva, ele adicionará um registro de <**ready** T> para o log e importará o log , importará o log (com todos os registros de T) à memória estável e enviará uma mensagem de **ready** T para C1

Commit Protocol em duas fases (Fase 1)



Commit Protocol em duas fases

- Fase 02
 - C1 – recebe de todos os sites mensagens de resposta ao **prepare T** ou é completado um intervalo de tempo prefixado.
 - Baseado nessas informações ele determina se a transação deve ser:
 - Efetivada – se C1 receber uma mensagem **ready T** de todos os sites participantes.
 - Será acrescentado um registro <**commit T**>
 - Abortado – se C1 receber pelo menos 1 mensagem de **abort T** de um dos sites participantes.
 - Será acrescentado um registro <**abort T**>
 - C1 – Mandará uma mensagem de commit ou abort a todos os sites participantes.

Commit Protocol em duas fases (Fase 2)



Log:
<prepara T>
<commit T>
<complete T>

Log:
<ready T>
<commit T>

<ready T>
<abort T>

<commit T>
<acknowledge T>

Manuseio de falhas

- Falha de um site participante
 - Ações do coordenador
 - Ações do Site após o retorno
 - Registros do log: commit, abort, ready, nada
- Falha no coordenador
 - Ações dos Sites
 - Ações do coordenador após o retorno
- Particionamento da rede

A maior desvantagem do protocolo 2PC é que uma falha do coordenador pode acabar em obstrução, que implica postergar a decisão de efetivar ou abortar T até que o coordenador volte a funcionar

Recuperação e Controle de Concorrência do 2PC

Volta de S

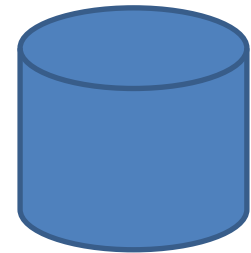


$T \rightarrow \text{In-doubt}$



`<ready T> ? <commit T> ||
<abort T>`

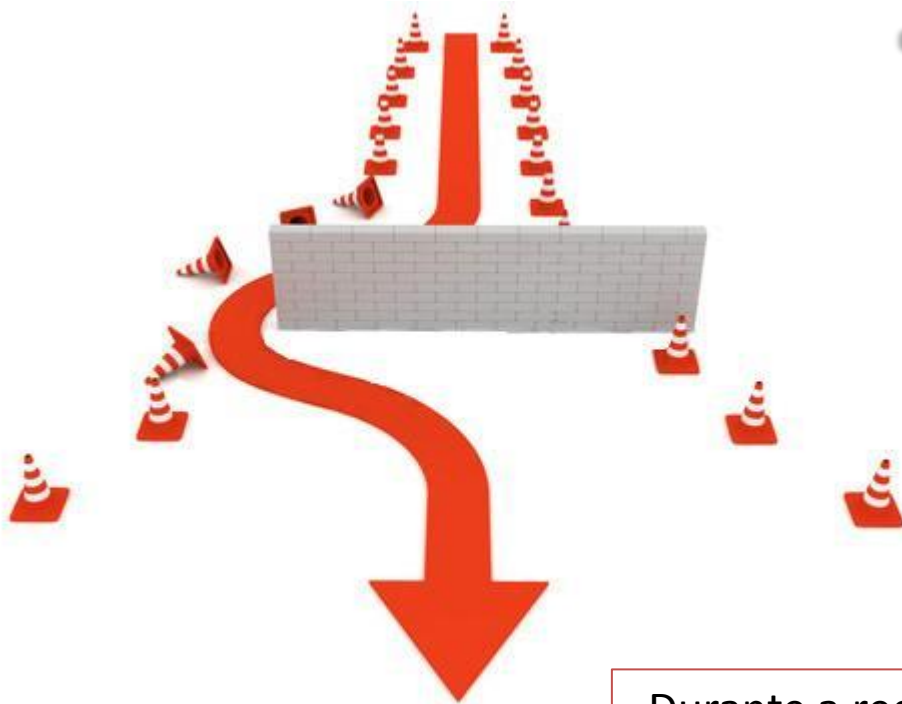
Volta a
processar!!



Recuperação e Controle de Concorrência do 2PC

- Descobrir o status das transações em dúvida pode ser demorado, pode ser necessário contatar diversos sites.
- Se o coordenador está fora do ar e nenhum outro site possui o status de uma transação incompleta, a recuperação fica **potencialmente obstruída caso se esteja usando o protocolo 2PC.**
 - Longo período ocioso!!!

Solucionado o problema



$\langle \text{ready } T \rangle \rightarrow$

$\langle \text{ready } T, L \rangle$

L é a lista de todos os bloqueios realizados pela transação T

Durante a recuperação, depois das ações locais de recuperação, para toda transação com dúvida, todos os bloqueios notificados (L) são reassumidos.

Commit protocol em três fases

- Premissas
 - Não ocorra particionamento de rede
 - No máximo K participantes por transação podem falhar enquanto o protocolo estiver sendo executado. K é um parâmetro indicativo da tolerância a falhas do protocolo nos sites
 - Em um tempo qualquer, ao menos $K + 1$ sites precisam estar ativos.

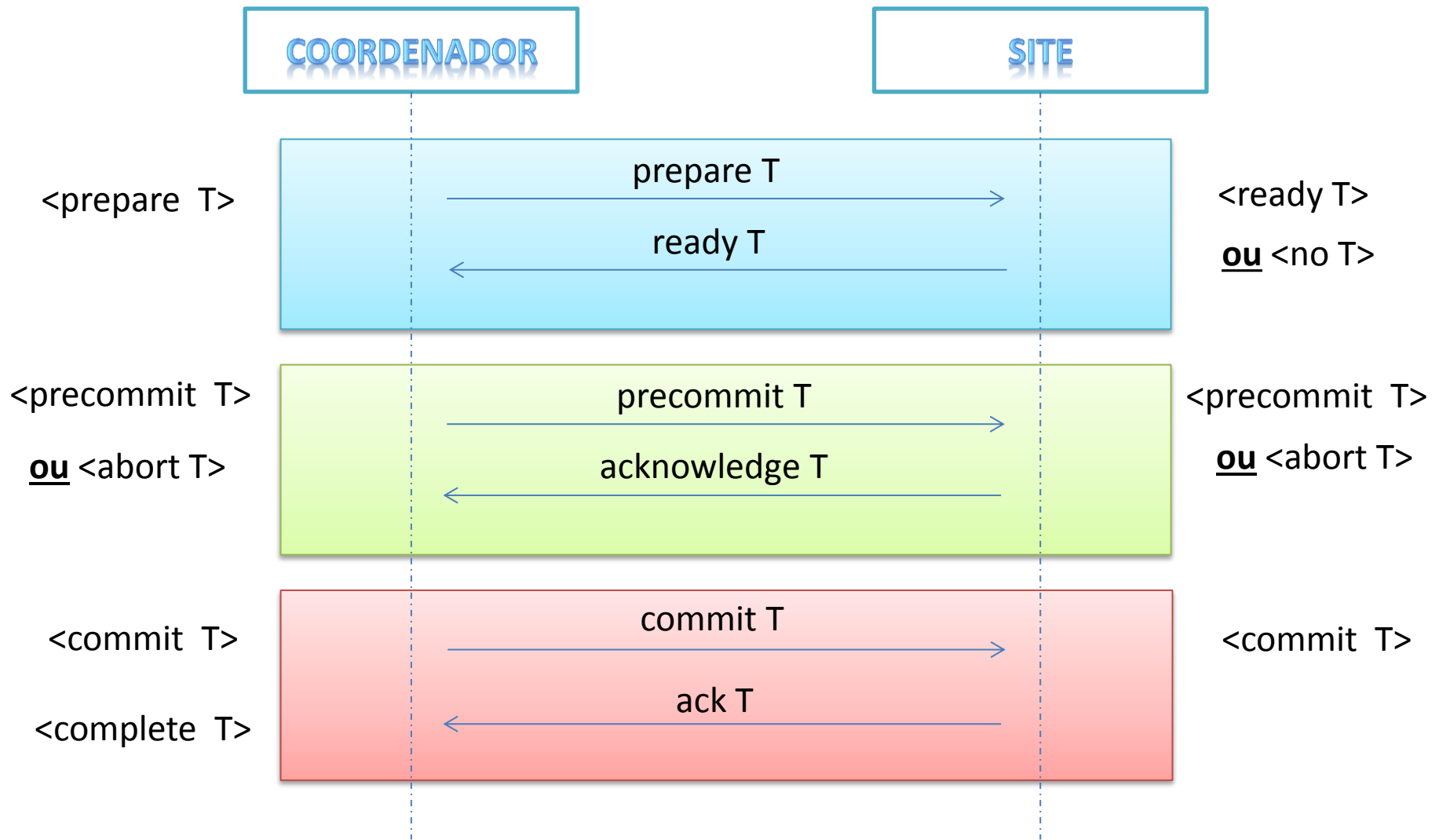
Commit protocol em três fases

- Fase 1. Idêntica a fase 1 do protocolo 2PC
- Fase 2.
 - C1 – Se não receber respostas de um site participante ou receber um **abort T**, então optará pelo **abort T**
 - C1 – Se recebe um **ready T** de todos os sites
 - LOG: *<precommit T>* - ainda pode ser abortada!
 - MSG: precommit T
 - S1 – Recebe o **precommit T**
 - LOG: *<precommit T>*
 - MSG: acknowledge T

Commit protocol em três fases

- Fase 3.
 - Esta fase é executada somente se na fase 2 for tomada a decisão de pré-efetivação
 - C1 – aguarda pelo menos K mensagens acknowledge
 - Toma a decisão em prol da efetivação
 - LOG: <commit T>
 - MSG: <commit T> a todos os sites participantes
 - S1 – recebe a mensagem
 - LOG: <commit T>

Commit em três fases



Manuseio de falhas

- Falha em um site participante
 - Depende do que está escrito no log
- Falha no coordenador
 - Como não consideramos particionamento de rede a única falha possível é no coordenador.

Protocolo de Falha do Coordenador

- O protocolo de falha do coordenador é ativado por um site participante à medida que não recebe nenhuma resposta dentro de determinado intervalo de tempo.
 1. Os sites participantes elegem um novo coordenador usando um protocolo de eleição
 2. O novo coordenador, C_{novo} , manda uma mensagem para cada site participante, solicitando o status local de T .
 3. Cada site participante. Inclusive C_{novo} , determina o status local de T^*
 4. Dependendo da resposta recebida, C_{novo} decide se efetiva ou aborta T , ou se reinicia o protocolo 3 PC

Questão 41. CESGRANRIO - 2010 - EPE - Analista de Gestão Corporativa - Tecnologia da Informação

[41] Em determinada funcionalidade de um sistema interno de um órgão público, é necessário realizar uma transação em um banco de dados distribuído, segundo o protocolo Two-Phase Commit (2PC) ou o Three-Phase Commit (3PC), sobre os quais afirma-se que

- (A) ambos são bloqueantes.
- (B) existem dois nós coordenadores no 2PC.
- (C) na fase de expansão, ambos adquirem write-lock.
- (D) na fase de retração, o 3PC adquire write-lock.
- (E) somente o 2PC é bloqueante.

Questão 42. CESGRANRIO - 2009 - Casa da Moeda - Analista de Nível Superior - Banco de Dados

[25] Em determinada reunião técnica sobre a construção de um sistema de informação distribuído, foram feitas as afirmativas a seguir sobre o protocolo two-phase commit.

I – Uma transação somente é confirmada (committed) se todos os participantes concordarem e estiverem prontos para realizar o commit.

II – O protocolo garante que todos os participantes da transação distribuída podem confirmar ou abortar seu trabalho.

III – Durante a segunda fase, o coordenador informa a todos os participantes sobre a decisão e confirma ou aborta a transação.

Está(ão) correta(s) a(s) afirmativa(s)

(A) I, apenas. (B) II, apenas. (C) III, apenas. (D) I e II, apenas.
(E) I, II e III.

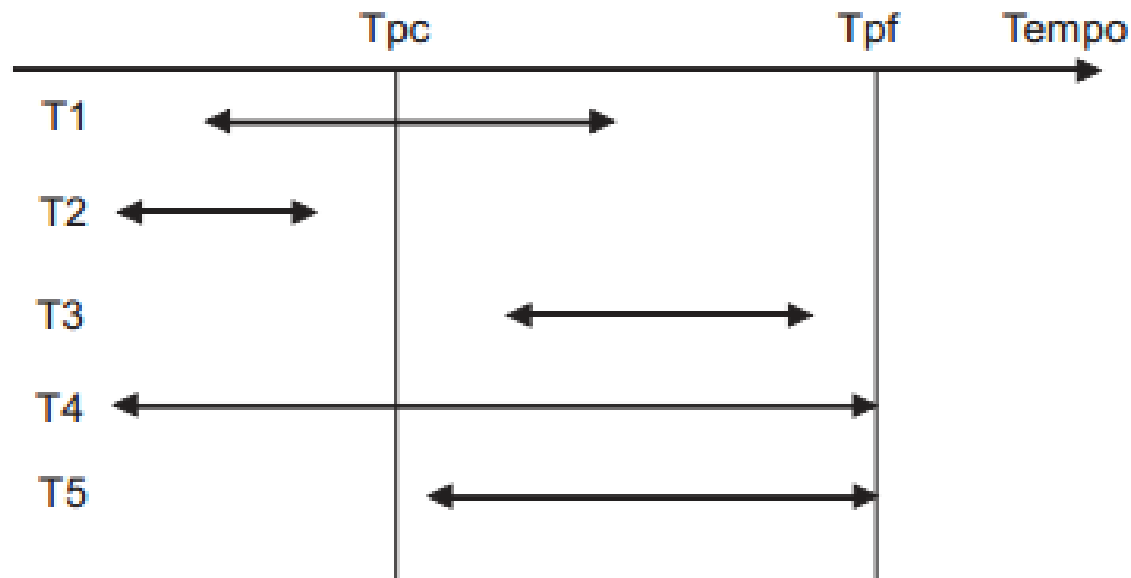
Questão 43. CESGRANRIO - 2008 - BNDES - Profissional Básico - Análise de Sistemas - Suporte

[69] No âmbito de bancos de dados distribuídos, seja T uma transação distribuída que deve ser efetivada (committed) segundo o protocolo de compromissamento de duas fases (two-phase commit). Sobre esse assunto, assinale a afirmativa INCORRETA.

- (A) Para que T seja efetivada (commit), todos os nós participantes de T devem, necessariamente, concordar com a efetivação.
- (B) O coordenador de T pode, incondicionalmente, abortá-la.
- (C) Na fase 1, o coordenador de T envia uma mensagem <prepare T> para todos os participantes de T, onde cada nó envia sua resposta a todos os outros nós.
- (D) Quando um nó participante de T e não coordenador recupera-se de uma falha e seu log apresenta um registro <commit T>, o nó deve executar redo(T).
- (E) Quando um nó participante de T e não coordenador recupera-se de uma falha e seu log apresenta um registro <abort T>, o nó deve executar undo(T).

Questão 44. CESGRANRIO - 2010 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

- [49] A figura abaixo apresenta a execução de cinco tipos de transações ao longo do tempo em um sistema gerenciador de bancos de dados cujas transações apresentam as propriedades ACID. No tempo Tpc é marcado um checkpoint, em Tpf há uma falha de sistema, as transações do tipo T1, T2 e T3 são concluídas com sucesso e as transações do tipo T4 e T5 foram iniciadas, mas não foram concluídas até o instante Tpf.



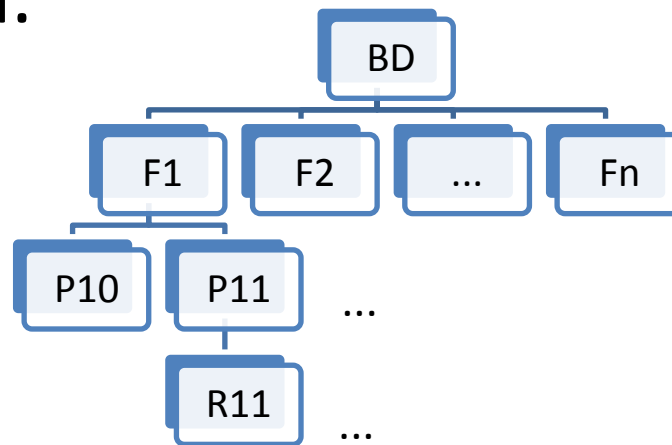
Questão 44. CESGRANRIO - 2010 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

Após a falha ocorrer, o sistema é reinicializado e passa por um processo de recuperação até atingir um estado correto. No contexto apresentado, somente devem ser desfeitas as transações

- (A) T1 e T3.
- (B) T4 e T5.
- (C) T1, T2 e T3.
- (D) T1, T3 e as transações concluídas sem sucesso antes do instante T_{pf}.
- (E) T4, T5 e as transações concluídas sem sucesso antes do instante T_{pf}.

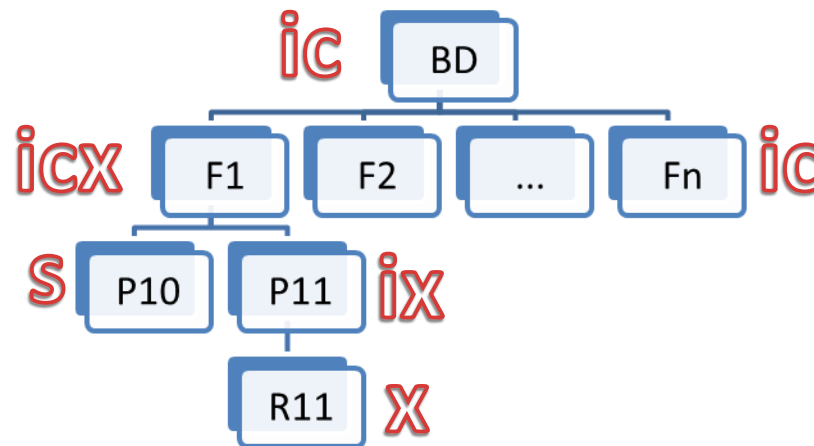
Bloqueio de granularidade múltipla (MGL)

- O tamanho dos itens de dados é chamado de granularidade do item de dados.
 - A granularidade fina refere-se a tamanho pequeno de itens, enquanto a granularidade grossa refere-se aos tamanhos grandes.
- Quanto menor o item bloqueado maior a concorrência.



Bloqueio de granularidade múltipla (MGL)

- São necessários bloqueios adicionais, chamados de bloqueios de intenção:*
- 1. Intenção compartilhada (IS ou IC)
- 2. Intenção exclusiva (IX)
- 3. Compartilhada-intenção-exclusiva (SIX ou ICX)



Questão 45. CESGRANRIO - 2010 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

[51] Considere os seguintes tipos de bloqueio utilizados em um sistema de banco de dados, sendo T uma transação que solicitou um bloqueio do tipo indicado sobre a RelVar (variável relacional) R:

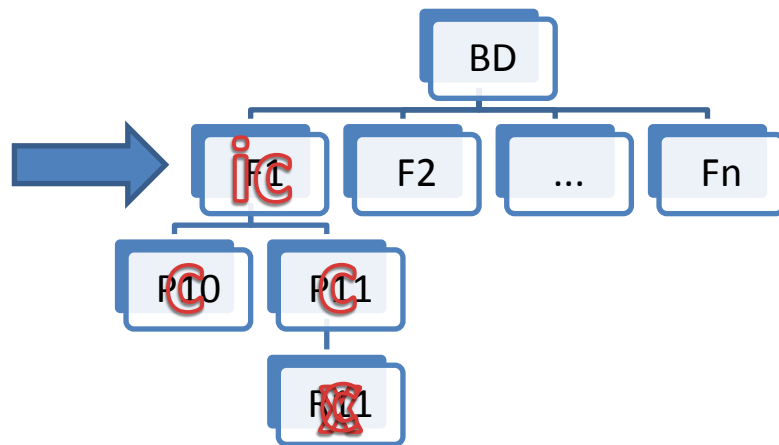
- IC: T tem a intenção de definir bloqueios do tipo C sobre tuplas individuais em R, a fim de garantir a estabilidade dessas tuplas enquanto elas estiverem sendo processadas.
- IX: Igual a IC, além disso, T poderia atualizar tuplas individuais em R e, dessa forma, definir bloqueios do tipo X sobre essas tuplas.
- C: T pode tolerar leitores concorrentes, mas não atualizadores concorrentes em R (a própria T não atualizará quaisquer tuplas em R).
- ICX: Combina C e IX, isto é, T pode tolerar leitores concorrentes, mas não atualizadores concorrentes em R. Além disso, T poderia atualizar tuplas individuais em R e, por isso, definir bloqueios do tipo X sobre essas tuplas.
- X: T não pode tolerar qualquer acesso concorrente a R. A própria T poderia ou não atualizar tuplas individuais em R.

Matriz de compatibilidade de bloqueio para bloqueio de granularidade múltipla

Transação t1

Transação t2

	IC	IX	C	ICX	X
IC	SIM	SIM	SIM	SIM	NÃO
IX	SIM	SIM	NÃO	NÃO	NÃO
C	SIM	NÃO	SIM	NÃO	NÃO
ICX	SIM	NÃO	NÃO	NÃO	NÃO
X	NÃO	NÃO	NÃO	NÃO	NÃO



ic → ic | ix | c | icx

ix → ic | ix

Questão 45. CESGRANRIO - 2010 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

Se uma transação A mantém um determinado tipo de bloqueio sobre uma RelVar R qualquer e uma transação distinta B emite uma requisição de bloqueio sobre R que não pode ser satisfeita imediatamente, temos um conflito; caso contrário, temos uma compatibilidade. Uma compatibilidade ocorre, portanto, quando A mantém um bloqueio do tipo

- (A) C sobre R e B emite uma requisição do tipo IX.
- (B) X sobre R e B emite uma requisição do tipo IC.
- (C) IC sobre R e B emite uma requisição do tipo ICX.
- (D) IX sobre R e B emite uma requisição do tipo C.
- (E) ICX sobre R e B emite uma requisição do tipo X.

Questão 46. CESGRANRIO - 2008 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

[57] Considere as duas transações (T1 e T2) de banco de dados abaixo

T1	T2
read_lock(Y);	read_lock(X);
read_item(Y);	read_item(X);
write_lock(X);	write_lock(Y);
unlock(Y);	unlock(X);
read_item(X);	read_item(Y);
$X := X + Y;$	$Y := Y + X;$
write_item(X);	write_item(Y);
unlock(X);	unlock(Y);

read_lock – implementa bloqueio compartilhado no item de banco de dados

write_lock – implementa bloqueio exclusivo no item de banco de dados

unlock – desbloqueia o item de banco de dados

read_item – executa a leitura do item

write_item – executa a escrita do item

Questão 46. CESGRANRIO - 2008 - Petrobrás - Analista de Sistemas Júnior - Infra-Estrutura

É correto afirmar que estas transações

- (A) seguem o protocolo de bloqueio em duas fases e não entram em impasse (deadlock), se executadas concorrentemente.
- (B) podem entrar em impasse (deadlock), se executadas concorrentemente.
- (C) são serializáveis e não seguem o protocolo de bloqueio em duas fases.
- (D) não são serializáveis e, portanto, podem ser executadas concorrentemente.
- (E) não seguem o protocolo de bloqueio em duas fases, não sendo portanto serializáveis.

Questão 47. CESGRANRIO - 2009 - Casa da Moeda - Analista de Nível Superior – Des. de Sistemas

[14] Em determinada transação de um sistema de contabilidade, as chaves estrangeiras devem ser avaliadas somente no COMMIT dessa transação. Que propriedade pode ser aplicada em uma restrição (constraint) para atingir esse comportamento?

- (A) DEFERRABLE
- (B) NO INDEX
- (C) NOW
- (D) ABORT
- (E) CASCADE

DEFERRABLE CONSTRAINTS

- São úteis em casos tais como
 1. **Na atualização de 2 tabelas que possuem relacionamento**, onde precisamos incluir uma linha nova, primeiro na tabela que contém a **chave-estrangeira** (tabela filha) e depois na tabela que contém a **chave-primária** do relacionamento (tabela mãe).
 2. **Na inserção de dados em massa**. Ao submeter comandos para inserir, por exemplo, **milhares de linhas em uma tabela**, o processo de inserção ocorre mais rápido se a checagem das constraints for executada somente no final, de uma única vez.

Exemplo

- CREATE TABLE EMPREGADO (
 ID NUMBER,
 NOME VARCHAR2(30),
 DEPARTAMENTO_ID NUMBER,
 CONSTRAINT EMP_PK PRIMARY KEY
 (ID),
 CONSTRAINT EMP_DEPTO_FK FOREIGN
 KEY (DEPARTAMENTO_ID)
 REFERENCES DEPARTAMENTO
 (DEPARTAMENTO_ID)
 DEFERRABLE INITIALLY DEFERRED);



Comportamento do Deferrable

- A qualquer momento é possível alterar o "comportamento" DEFERRABLE da CONSTRAINT:
- Se for desejado alterar para validar imediatamente (sem postergação)
 - SET CONSTRAINTS EMP_DEPTO_FK IMMEDIATE;
- Se for desejado alterar para validar somente no final da transação (com postergação):
 - SET CONSTRAINTS EMP_DEPTO_FK DEFERRED;

Questão 48. CESGRANRIO - 2005 - AL-TO - Assistente Legislativo – Esp. - Programação de Computadores

[38] Um SGBD para manter a integridade dos dados deve apresentar algumas propriedades para as transações. A propriedade que define “ou todas as operações da transação são refletidas corretamente no banco de dados ou nenhuma deve ser refletida” é:

- (A) atomicidade.
- (B) consistência.
- (C) durabilidade.
- (D) isolamento.
- (E) polimorfismo.

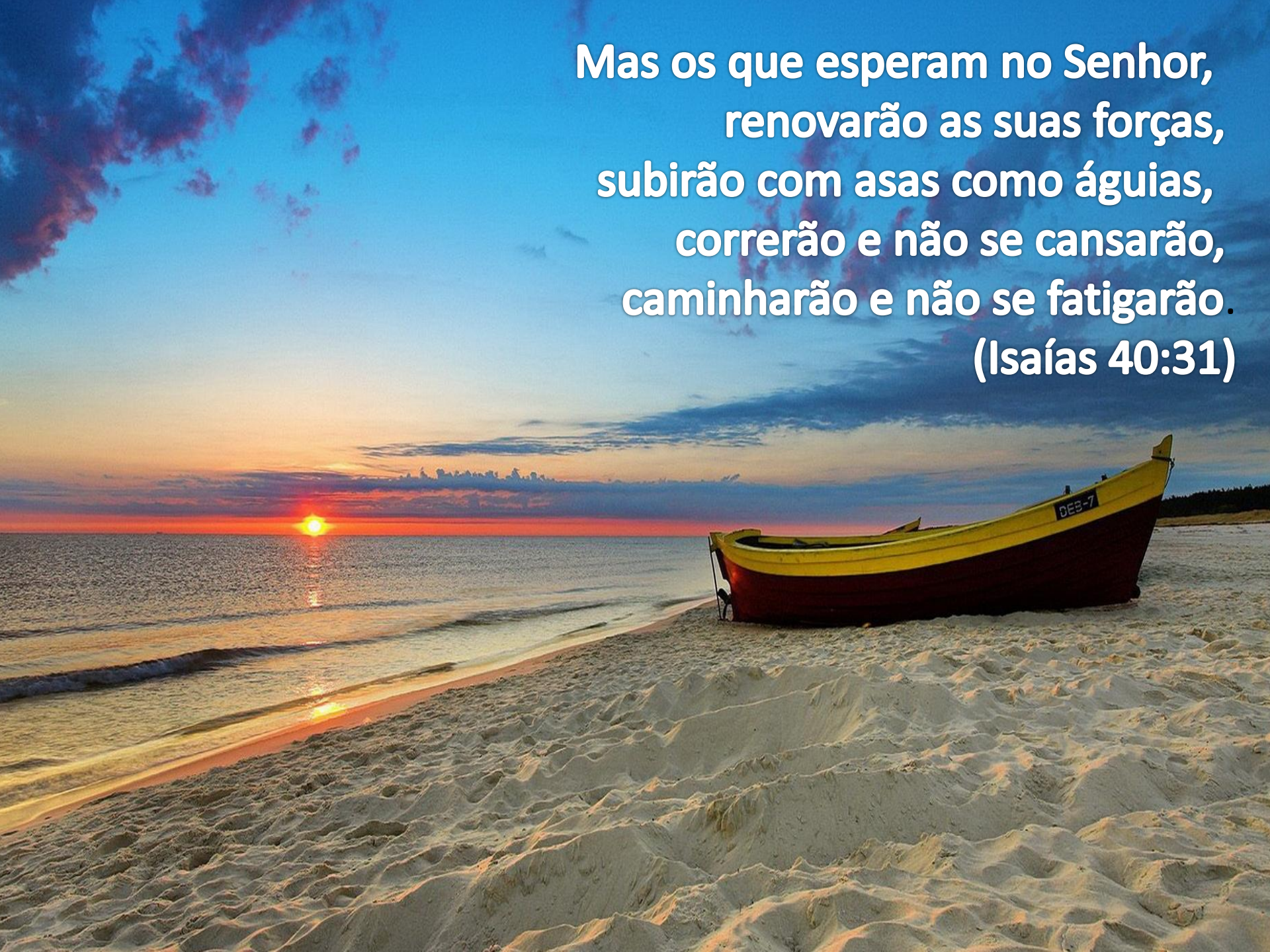


rcthiago@gmail.com

OBRIGADO! Até o próximo módulo!



**Mas os que esperam no Senhor,
renovarão as suas forças,
subirão com asas como águias,
correrão e não se cansarão,
caminharão e não se fatigarão.
(Isaías 40:31)**



Banco de dados

Módulo 02 – Transações, Concorrência e Recuperação

Curso Preparatório - ITnerante

Prof. Thiago Cavalcanti



Gabarito

- | | | |
|-----------------------------|----------------------|------------|
| 1. [107] F | 19. [128] C | 37. [69] B |
| 2. [60] A | 20. [92] C | 38. [47] E |
| 3. [66] C | 21. [67] E [68] E | 39. [41] A |
| 4. [22] A | 22. [88] C | 40. [47] E |
| 5. [106] V | 23. [85] C [86] C(?) | 41. [41] E |
| 6. [129] E | 24. [87] E [88] C | 42. [14] A |
| 7. [54] C | 25. [94] E | 43. [25] C |
| 8. [70] F | 26. [50] D | 44. [49] B |
| 9. [85] V | 27. [79] c | 45. [51] C |
| 10. [86] F | 28. [80] c | 46. [57] B |
| 11. [130] V | 29. [76] C | 47. [69] A |
| 12. [78] D | 30. [77] D | 48. [48] A |
| 13. [100] E [101] E [102] C | 31. [50] B | |
| 14. [87] E | 32. [49] D | |
| 15. [89] E [101] C | 33. [15] E | |
| 16. [86] C | 34. [21] D | |
| 17. [126] C | 35. [41] D | |
| 18. [127] C | 36. [58] C | |

Estratégias para liberação de um item de dado modificado da “cache” para o disco.

- **Atualização in-place:** grava o item no mesmo local do disco, apagando o valor anterior, mantendo apenas uma cópia do dado
 - É necessário Log gravado antes do BD
- **“Shadowing” (imagem):** grava o novo item em uma localização diferente, mantendo pelo menos duas cópias do item de dado
 - Não é necessário o Log
- **Write-Ahead Logging:** na atualização inplace, garante que a ImAn seja gravada no log antes da gravação da ImDp no disco.