

# BANCO DE DADOS Transações e Técnicas de Recuperação

Prof. Eduardo Neves  
[edumneves@gmail.com](mailto:edumneves@gmail.com)

Dúvidas de Banco de Dados:  
Fórum do curso 😊

Artigos do Blog:  
<http://goo.gl/Sm6JgV>

# BD – Distribuição das aulas

- ▶ Aula 01
  - Introdução
  - Transações – COMMIT, ROLLBACK
- ▶ Aula 02
  - Propriedades ACID
- ▶ Aula 03
  - Exemplos e Exercícios
- ▶ Aula 04
  - Escalonamento de Transações
- ▶ Aula 05
  - Níveis de Isolamento

# BD – Distribuição das aulas

- ▶ Aula 06
  - Bateria de questões
- ▶ Aula 07
  - Equivalência por conflito
- ▶ Aula 08
  - Serialização por conflito
- ▶ Aula 09
  - Equivalência de visão
- ▶ Aula 10
  - Recuperação e log

# BD – Distribuição das aulas

- ▶ Aula 11
  - Atualização adiada – NO-UNDO/REDO
- ▶ Aula 12
  - Atualização imediata – UNDO/REDO
- ▶ Aula 13
  - Bateria de questões
- ▶ Aula 14
  - Paginação sombra – NO-UNDO/NO-REDO
- ▶ Aula 15
  - Bateria de questões

E

Eduardo Neves

Inscrever-se 66

Envios Reproduzir todos

**Pegadinha da Semana - 4 - FUNDATEC SQL LEFT JOIN**  
29 visualizações • 1 dia atrás

**Revisão Anki - Redes para concursos**  
185 visualizações • 2 semanas atrás

**Pegadinha da Semana - 3 - FGV GROUP BY HAVING**  
103 visualizações • 3 semanas atrás

**Pegadinha da semana - SQL Exists - parte 2**  
36 visualizações • 3 semanas atrás

**Pegadinha da semana - SQL Exists - parte 1**  
17 visualizações • 3 semanas atrás

Playlists criadas

**Pegadinha da Semana**  
Atualizada há 2 dias

**Anki**

# Dúvidas de Banco de Dados

## Fórum Geral de Dúvidas

Fórum Geral de Dúvidas

Acrescentar um novo tópico de discussão

Tópico	Autor	Comentários	Última mensagem
Dúvida Questão Normalização FNBC	 Danniel Albuquerque Araújo	3	Eduardo Neves Dom, 14 Ago 2016, 07:41
Dúvida Questão FGV sobre normalização FNBC	 Danniel Albuquerque Araújo	3	Eduardo Neves Dom, 14 Ago 2016, 07:40
Dúvida questão	 André Felipe Mendonça Andrade	3	Eduardo Neves Dom, 14 Ago 2016, 07:40
Dúvida questão	 Gustavo Mantuan	2	Gustavo Mantuan Dom, 10 Abr 2016, 10:24
Dúvida Aula 5 Questão 32 fgv	 Gustavo Mantuan	3	Eduardo Neves Sáb, 9 Abr 2016, 12:36

# Gostou? Avalie o curso 😊

## **Excelente** Comentado por DHEISON

Qualidade ★★★★★

Preço ★★★★★

Conteúdo ★★★★★

Professor ★★★★★

Após ver aulas de 3 professores diferentes tava começando a achar que o problema era comigo, mas finalmente uma boa didática para entender 100% o conteúdo proposto, recomendado! (Postado em 24/02/16)

## **Excelente Curso de BD** Comentado por RAFAEL NOGUEIRA

Qualidade ★★★★★

Preço ★★★★★

Conteúdo ★★★★★

Professor ★★★★★

Curso muito didático com explicações na medida! :)

Melhor explicação que ja vi sobre conceitos confusos e abstratos como o ansi-spark e DBA vs AD!

5 Estrelas fácil! (Postado em 03/02/16)

## **Muito bom!** Comentado por Leonardo

Qualidade ★★★★★

Preço ★★★★★

Conteúdo ★★★★★

Professor ★★★★★

Excelente Professor!

Teoria na medida certa e muitos exercícios!

Só em Arquitetura ANSI/SPARC foram mais de 30 exercícios!

Recomendado!

Grande Abraço! (Postado em 03/11/15)

- ▶ Combo de SQL
- ▶ <https://goo.gl/YwK6fj>

## Combo SQL - Prof. Eduardo Neves

Dê Sua Opinião Sobre o Produto

BANCO DE  
DADOS

COMBO - SQL

Cursos de BD SQL do Prof. Eduardo Neves



SISTEMAS

Prof. Eduardo Neves  
COMBO SQL



**Atenção:** Este Combo é uma coletânea de nossos módulos (não é um curso específico). Você pode selecionar os itens que deseja e o valor será recalculado a cada item. Lembre-se de que nosso **desconto progressivo** é **aplicado no carrinho**. O valor pode cair em quase **40%**. Confira!!!

🛒 Adicionar ao Carrinho

👍 Curtir 5

👍 G+1 0

🐦 Tweetar

📌 Compartilhar 1

DESCRIÇÃO DO MATERIAL

INFORMAÇÕES ADICIONAIS

# Apresentação

## ▶ Formação

- Bacharel em Ciência da Computação/UFRJ
- Pós-graduação – Gestão de Projetos

## ▶ Analista de Sistemas – BNDES – 2013

## ▶ Principais aprovações

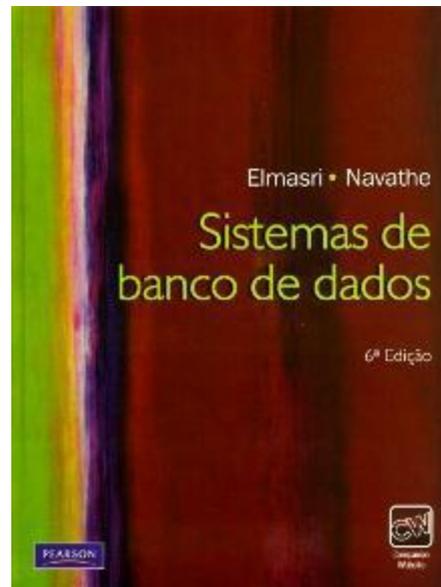
- 4º – BNDES/2012 – CESGRANRIO
- 19º – BNDES/2011 – CESGRANRIO
- 30º – Petrobras/ 2011 – CESGRANRIO
- 12º – FINEP/2011 – CESGRANRIO
- 3º – Transpetro/2011 – CESGRANRIO
- 42º – TRT-RJ/2011 – FCC
- 7º – Petrobras Macaé/2010 – CESGRANRIO
- 1º – Caixa Econômica Federal/2010 – Nível médio informática – CESPE
- 18º – BR Distribuidora/2010 – Analista SAP – CESGRANRIO

# Dicas de estudo Gerais

- ▶ **NÃO DESISTIR!!!!**
- ▶ Aprender com os próprios erros
- ▶ Estudar as matérias em ciclos
  - <https://goo.gl/1sxeTE>
- ▶ Ter uma forma de revisão
  - Anki
    - <https://goo.gl/SP1tJt>
  - Mapa Mental
  - Resumos
- ▶ Fazer muitos exercícios
  - Da mesma banca primeiro da mais recente para a mais antiga
  - De outras bancas
  - Usar sites de questões
    - (gabaritou, qconcursos.com, mapadaprova, tecconcursos, ...)
- ▶ Timasters
  - Tirar dúvidas, compartilhar conhecimento.

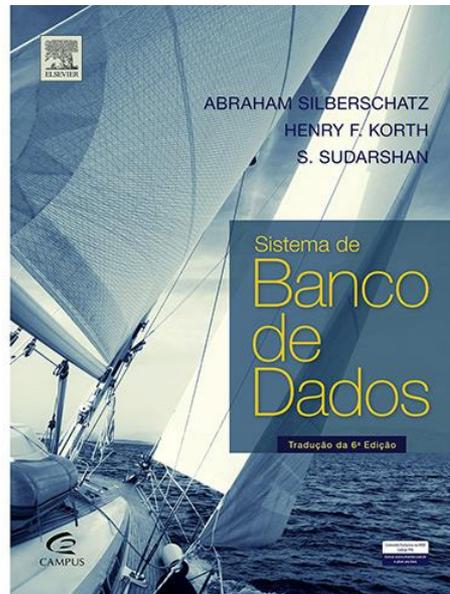
# Bibliografia

**Sistemas de Banco de Dados – 6ª edição**  
Ramez E. Elmasri, Shamkant B. Navathe  
Editora Pearson, 2011



# Bibliografia

**Sistemas de Banco de Dados – 6ª edição**  
**Abraham Silberschatz, Henry F. Korth**  
**Editora Campus, 2012**

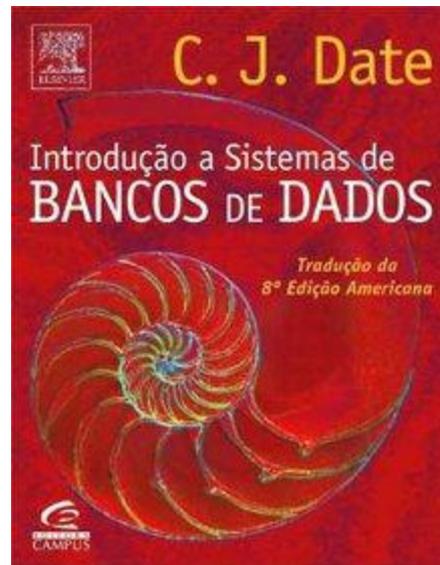


# Bibliografia

## Introdução a Sistemas de Banco de Dados

C.J. Date

Editora Campus, 2004



# BANCO DE DADOS

# Transações

Prof. Eduardo Neves  
[edumneves@gmail.com](mailto:edumneves@gmail.com)

Dúvidas de Banco de Dados:  
Fórum do curso 😊

Artigos do Blog:  
<http://goo.gl/Sm6JgV>

# Transações

- ▶ Unidade lógica de trabalho, constituída de um conjunto de operações, cujo objetivo é transformar um BD de um estado consistente para outro estado consistente, mesmo que nos passos intermediários o sistema permaneça temporariamente inconsistente.
  - ▶ Pode conter diversas operações (inclusão, exclusão, atualização e consulta)
- 

# Transações

## ▶ ANSI

- BEGIN TRANSACTION → Inicia a transação
- COMMIT → Conclui a transação, confirmando o trabalho.
- ROLLBACK → Aborta a transação, desfazendo o trabalho.

## ▶ SQL SERVER

- BEGIN TRANSACTION → Inicia a transação
  - END TRANSACTION → Termina a transação, se não deu erro confirma, se deu erro desfaz.
  - COMMIT e ROLLBACK normais
  - Autocommit habilitado por padrão.
- 

# Transações

## ▶ Oracle

- Uma transação começa quando for executada a 1ª instrução SQL executável
- COMMIT e ROLLBACK normais
- Autocommit desabilitado por padrão.

## ▶ MySQL

- START TRANSACTION ou BEGIN → Inicia a transação
- COMMIT e ROLLBACK normais
- Autocommit habilitado por padrão.

# Exemplo

```
BEGIN TRANSACTION
```

```
UPDATE Empregado SET Salario = Salario * 0.9  
WHERE Salario > 5000
```

```
IF @@ERROR <> 0
```

```
ROLLBACK
```

```
ELSE
```

```
COMMIT
```

# Operações Básicas

- ▶ Modelo simplificado para processamento de transações
- ▶ Operações Básicas:
  - Leitura de um item: **read\_item(x)** ou **ler\_item(x)** ou **r(x)**
    - Lê um item do banco de dados chamado X para uma variável do programa também chamada X.
  - Escrita de um item: **write\_item(x)** ou **gravar\_item(x)** ou **w(x)**
    - Grava o valor da variável de programa X no item de banco de dados chamado X.

# Operações Básicas

- ▶ Exemplo:

$T_1$
read_item( $X$ ); $X := X - N$ ; write_item( $X$ ); read_item( $Y$ ); $Y := Y + N$ ; write_item( $Y$ );

$T_2$
read_item( $X$ ); $X := X + M$ ; write_item( $X$ );

# Savepoint

```
BEGIN TRANSACTION
```

```
.....operações1
```

```
SAVEPOINT P1;
```

```
.....operações2
```

```
IF @Erros > 0
```

```
    ROLLBACK TO SAVEPOINT P1;
```

```
.....operações3
```

```
IF @Erros > 0
```

```
    COMMIT;
```

```
ELSE
```

```
    ROLLBACK;
```

- ▶ **RELEASE SAVEPOINT P1** (destruir um ponto de salvamento)

# Propriedades das Transações

- ▶ Requisitos que sempre devem ser atendidos por uma transação
- ▶ Propriedades **ACID**
  - **A**tomicidade
  - **C**onsistência
  - **I**solamento
  - **D**urabilidade

# Propriedades das Transações

## ▶ **Atomicidade:**

- uma transação é uma unidade de processamento, é realizada integralmente ou não é realizada.

## ▶ **Consistência:**

- uma transação leva um banco de dados de um estado consistente para outro estado consistente.

# Propriedades das Transações

## ▶ **Atomicidade:**

- uma transação é uma unidade de processamento, é realizada integralmente ou não é realizada.

## ▶ **Consistência:**

- uma transação leva um banco de dados de um estado consistente para outro estado consistente.

# Propriedades das Transações

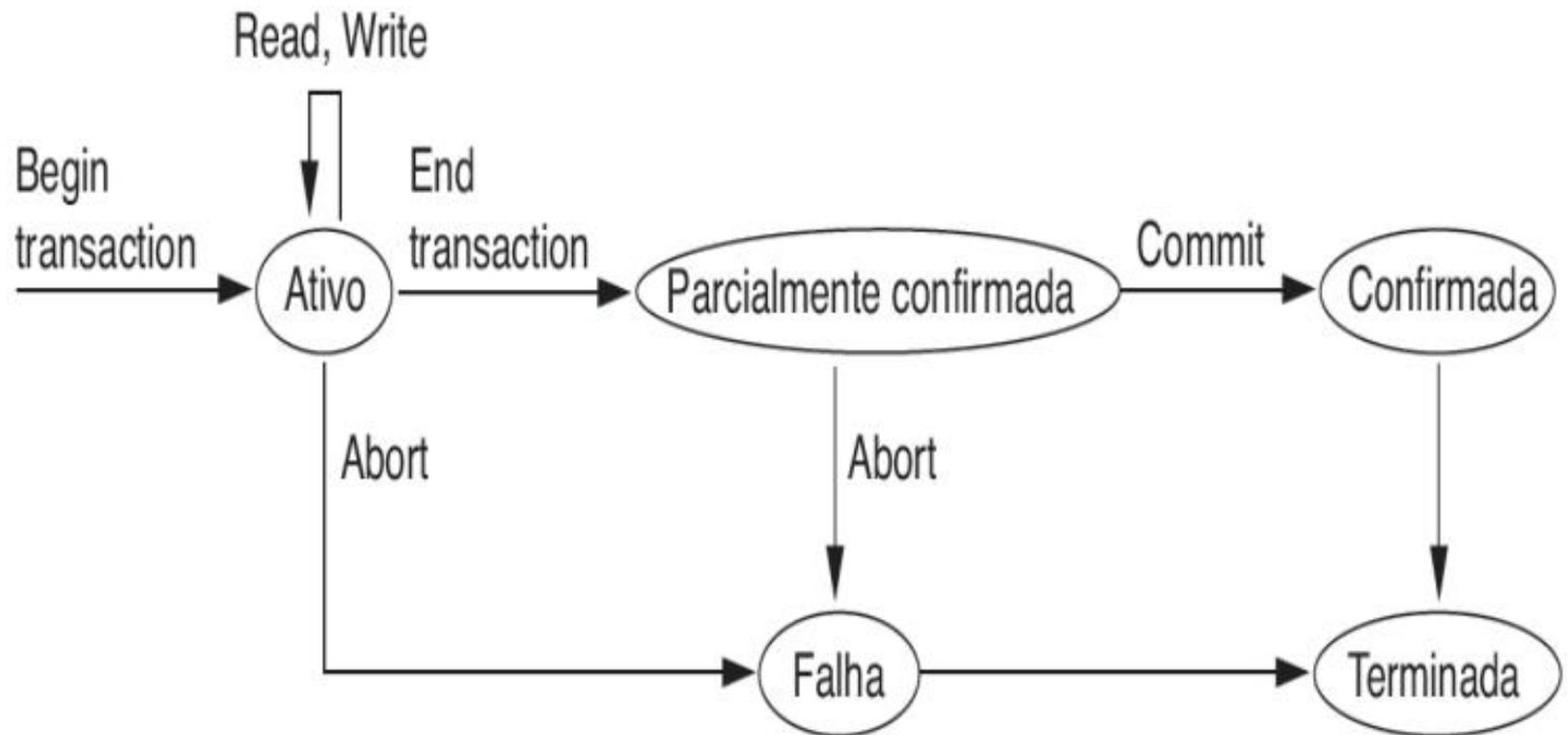
## ▶ Isolamento:

- uma transação deve parecer como se estivesse sendo executada isoladamente.

## ▶ Durabilidade:

- as alterações aplicadas a um banco de dados por meio de uma transação confirmada (committed) devem persistir no banco de dados.

# Estados de uma Transação



# Q1 – CESPE – TRT-17 – 2013

Em relação aos conceitos de transações, tipos de bancos de dados e índices, julgue os itens subsecutivos.

Para assegurar a integridade dos dados, exige-se que um sistema de banco de dados mantenha as seguintes propriedades das transações: atomicidade, consistência, isolamento e durabilidade.

# Q1 – CESPE – TRT-17 – 2013

Em relação aos conceitos de transações, tipos de bancos de dados e índices, julgue os itens subsecutivos.

**C** Para assegurar a integridade dos dados, exige-se que um sistema de banco de dados mantenha as seguintes propriedades das transações: atomicidade, consistência, isolamento e durabilidade.



# Q2 – CESPE – TJSE – 2014

Com relação a banco de dados, julgue o item que se segue.

A atomicidade de um SGBD garante que cada transação seja executada de maneira singular, ou seja, que cada transação possua um identificador unívoco. O isolamento do SGBD garante, por sua vez, que as transações sejam executadas isoladamente uma das outras.

# Q2 – CESPE – TJSE – 2014

Com relação a banco de dados, julgue o item que se segue.

**E** A atomicidade de um SGBD garante que cada transação seja executada de maneira singular, ou seja, que cada transação possua um identificador unívoco. O isolamento do SGBD garante, por sua vez, que as transações sejam executadas isoladamente uma das outras.



# Q3 – CESPE – ANP – 2013

Acerca do conceito de processamento de transações em um banco de dados, julgue os itens a seguir.

Para assegurar a integridade dos dados em uma transação, exige-se que os sistemas de banco de dados mantenham, durante a transação, a propriedade denominada atomicidade, em que a soma das entradas e saídas das transações são sempre constantes e os campos são indivisíveis.

# Q3 – CESPE – ANP – 2013

Acerca do conceito de processamento de transações em um banco de dados, julgue os itens a seguir.

**E** Para assegurar a integridade dos dados em uma transação, exige-se que os sistemas de banco de dados mantenham, durante a transação, a propriedade denominada atomicidade, em que a soma das entradas e saídas das transações são sempre constantes e os campos são indivisíveis.



# Q4 – CESPE – CRPM – 2013

No que se refere aos conceitos de administração e sistema de gerenciamento de banco de dados (SGBD), julgue os itens que se subsecutivos.

Um SGBD utiliza o conceito de atomicidade do registro, garantindo que, detectada uma falha na operação com o registro, os dados sejam salvos em seu último estado consistente, anterior à falha.

# Q4 – CESPE – CRPM – 2013

No que se refere aos conceitos de administração e sistema de gerenciamento de banco de dados (SGBD), julgue os itens que se subsecutivos.

**E** Um SGBD utiliza o conceito de atomicidade do registro, garantindo que, detectada uma falha na operação com o registro, os dados sejam salvos em seu último estado consistente, anterior à falha.

**Q5** – Considere uma transação no banco de dados de uma instituição financeira referente à transferência de valores da conta-corrente de um determinado cliente para a conta-corrente de outro cliente, mostrada no quadro a seguir.

Transação	
1	Ler (X);
2	$X = X - 50;$
3	Escrever (X);
4	Ler (Y);
5	$Y = Y + 50;$
6	Escrever (Y);

Caso ocorra uma falha no sistema logo após ter sido executada a operação de número 3 e antes de ter sido executada a operação de número 6 do quadro acima, e se o sistema não conseguir restabelecer o valor original de X, qual propriedade de transações foi violada?

- (A) Atomicidade
- (B) Bloqueio
- (C) Seriabilidade
- (D) Isolamento
- (E) Normalização

Transação	
1	Ler (X);
2	$X = X - 50;$
3	Escrever (X);
4	Ler (Y);
5	$Y = Y + 50;$
6	Escrever (Y);

Caso ocorra uma falha no sistema logo após ter sido executada a operação de número 3 e antes de ter sido executada a operação de número 6 do quadro acima, e se o sistema não conseguir restabelecer o valor original de X, qual propriedade de transações foi violada?

-  (A) Atomicidade
- (B) Bloqueio
- (C) Seriabilidade
- (D) Isolamento
- (E) Normalização

Transação	
1	Ler (X);
2	$X = X - 50;$
3	Escrever (X);
4	Ler (Y);
5	$Y = Y + 50;$
6	Escrever (Y);

# Q6 – ESAF – CGU – 2012

Uma das seguintes instruções SQL finaliza uma transação

- a) Stand-up work.
- b) Commit work.
- c) Complete work.
- d) Roll-and-stop work.
- e) Put-up work.

# Q6 – ESAF – CGU – 2012

Uma das seguintes instruções SQL finaliza uma transação

- a) Stand-up work.
-  b) Commit work.
- c) Complete work.
- d) Roll-and-stop work.
- e) Put-up work.

**Q7** – Para que seja garantida a integridade de dados, as propriedades de uma transação em um Sistema Gerenciador de Banco de Dados são

- (A) ativação, recuperação de falha, compartilhamento e capacidade para abortar.
- (B) compatibilidade, solicitação, espera e concessão.
- (C) detecção de erros, serialização, tratamento de bloqueios e ordenação.
- (D) escrita, leitura, validação e finalização.
- (E) isolamento, durabilidade, consistência e atomicidade.

**Q7** - Para que seja garantida a integridade de dados, as propriedades de uma transação em um Sistema Gerenciador de Banco de Dados são

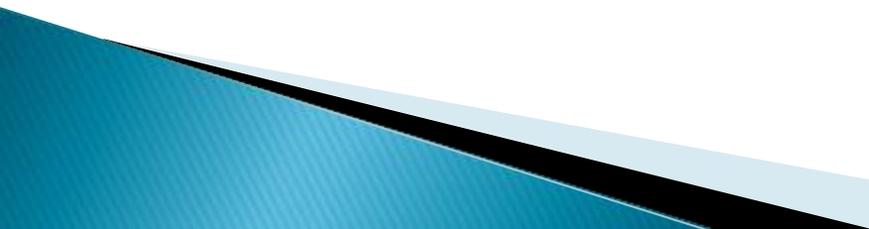
- (A) ativação, recuperação de falha, compartilhamento e capacidade para abortar.
- (B) compatibilidade, solicitação, espera e concessão.
- (C) detecção de erros, serialização, tratamento de bloqueios e ordenação.
- (D) escrita, leitura, validação e finalização.
- (E) isolamento, durabilidade, consistência e atomicidade.

# Q8 – FCC – TJPE – 2012

Considere:

- I. Se uma transação é concluída com sucesso (operação commit bem sucedida), então seus efeitos são persistentes.
- II. Ou todas as ações da transação acontecem, ou nenhuma delas acontece.

As propriedades (I) e (II) das transações em SGBDs, significam, respectivamente,

- a) durabilidade e consistência.
  - b) persistência e automação.
  - c) isolamento e atomicidade.
  - d) durabilidade e atomicidade.
  - e) consistência e persistência.
- 

# Q8 – FCC – TJPE – 2012

Considere:

- I. Se uma transação é concluída com sucesso (operação commit bem sucedida), então seus efeitos são persistentes.
- II. Ou todas as ações da transação acontecem, ou nenhuma delas acontece.

As propriedades (I) e (II) das transações em SGBDs, significam, respectivamente,

- a) durabilidade e consistência.
- b) persistência e automação.
- c) isolamento e atomicidade.
-  d) durabilidade e atomicidade.
- e) consistência e persistência.

# Q9 – FCC – MPEMA – 2013

O conceito de transações em um banco de dados relacional envolve algumas propriedades conhecidas e agrupadas sob o acrônimo ACID. A letra D nesse acrônimo significa Durabilidade, sendo que seu conceito indica que

- a) todas as regras de consistência previamente programadas no banco de dados serão automaticamente excluídas.
- b) cada transação é independente das demais, não gerando influência nas demais transações eventualmente existentes.
- c) os tipos de dados definidos não dependem do sistema gerenciador de banco de dados utilizado.
- d) ao término de uma transação, eventuais alterações em valores das tabelas do banco de dados devem persistir, mesmo que ocorram falhas no sistema de banco de dados.
- e) é necessário utilizar um sistema gerenciador de banco de dados capaz de implementar mecanismos de controle de concorrência.

# Q9 – FCC – MPEMA – 2013

O conceito de transações em um banco de dados relacional envolve algumas propriedades conhecidas e agrupadas sob o acrônimo ACID. A letra D nesse acrônimo significa Durabilidade, sendo que seu conceito indica que

- a) todas as regras de consistência previamente programadas no banco de dados serão automaticamente excluídas.
- b) cada transação é independente das demais, não gerando influência nas demais transações eventualmente existentes.
- c) os tipos de dados definidos não dependem do sistema gerenciador de banco de dados utilizado.
-  d) ao término de uma transação, eventuais alterações em valores das tabelas do banco de dados devem persistir, mesmo que ocorram falhas no sistema de banco de dados.
- e) é necessário utilizar um sistema gerenciador de banco de dados capaz de implementar mecanismos de controle de concorrência.

# Escalonamento de Transações

- ▶ Quando transações estão sendo executadas concorrentemente e de modo entrelaçado, a ordem de execução das operações das várias transações é conhecida como escalonamento (schedule).
- ▶ Escalonamentos:
  - Seriais
  - Concorrentes

# Escalonamento

- T1:

- $R_1(x)$
- $W_1(x)$
- $R_1(y)$
- $W_1(y)$

- T2:

- $R_2(x)$
- $W_2(x)$
- $R_2(y)$
- $W_2(y)$

- Escalonamento Serial:

- $R_1(x); W_1(x); R_1(y); W_1(y); R_2(x); W_2(x); R_2(y); W_2(y);$
- $R_2(x); W_2(x); R_2(y); W_2(y); R_1(x); W_1(x); R_1(y); W_1(y);$

- Escalonamento Concorrente (intercalado):

- $R_1(x); R_2(x); W_2(x); W_1(x); R_2(y); R_1(y); W_1(y); W_2(y);$
- $R_2(x); R_1(x); W_2(x); W_1(x); R_2(y); R_1(y); W_2(y); W_1(y);$

# Escalonamento Serializável

- ▶ Escala concorrente serializável
  - Sua execução equivale a alguma execução serial
  - Se a escala é serializável então ela é correta.

$T_1$	$T_2$
read (X);	read (X);
X:=X-N;	X := X*0,9;
write (X);	write (X);
read (Y);	
Y:=Y+N;	
write (Y);	

# Escalonamento Serial T1 → T2

T <sub>1</sub>	T <sub>2</sub>
read (X); X:=X-N; write (X); read (Y); Y:=Y+N; write (Y);	read (X); X := X*0,9; write (X);

Entrada:

X = 100

Y = 200

N = 5

Saída:

X = 85,5

Y = 205

# Escalonamento Serial T2 → T1

<u>T<sub>1</sub></u>	<u>T<sub>2</sub></u>
	read (X);
	X := X*0,9;
	write (X);
read (X);	
X:=X-N;	
write (X);	
read (Y);	
Y:=Y+N;	
write (Y);	

Entrada:

X = 100

Y = 200

N = 5

Saída:

X = 85

Y = 205

# Escalonamento Concorrente

T <sub>1</sub>	T <sub>2</sub>
read (X); X:=X-N; write (X);	read (X); X := X*0,9; write (X);
read (Y); Y:=Y+N; write (Y);	

Entrada:

X = 100

Y = 200

N = 5

Saída:

X = 85,5

Y = 205

Estado igual ao escalonamento

Serial T1 → T2.

Esse escalonamento é **Serializável**

# Escalonamento Concorrente

$T_1$	$T_2$
read (X); X:=X-N;	read (X); X := X*0,9;
write (X); read (Y);	write (X);
Y:=Y+N; write (Y);	

Entrada:

X = 100

Y = 200

N = 5

Saída:

X = 90

Y = 205

Esse escalonamento não é **Serializável**.

# Q10 – UEAP – CS–UFG – 2014

Serialização é um conceito pertinente ao isolamento de transações no controle de concorrência do banco de dados. Um escalonamento não serial  $S$  de um conjunto de  $n$  transações é serializável se  $S$  produzir o mesmo

- a) número de operações de algum escalonamento serial das mesmas  $n$  transações.
- b) número de operações de algum escalonamento não serial das mesmas  $n$  transações.
- c) estado final do banco de dados de algum escalonamento serial das mesmas  $n$  transações.
- d) estado final do banco de dados de algum escalonamento não serial das mesmas  $n$  transações.

# Q10 – UEAP – CS–UFG – 2014

Serialização é um conceito pertinente ao isolamento de transações no controle de concorrência do banco de dados. Um escalonamento não serial  $S$  de um conjunto de  $n$  transações é serializável se  $S$  produzir o mesmo

- a) número de operações de algum escalonamento serial das mesmas  $n$  transações.
- b) número de operações de algum escalonamento não serial das mesmas  $n$  transações.
-  c) estado final do banco de dados de algum escalonamento serial das mesmas  $n$  transações.
- d) estado final do banco de dados de algum escalonamento não serial das mesmas  $n$  transações.

# Níveis de isolamento

- ▶ As transações especificam um nível de isolamento
- ▶ Define o grau em que uma transação deve ser isolada contra modificações de recursos ou de dados feitas por outras transações.
- ▶ Os níveis de isolamento são descritos em termos de quais efeitos colaterais de simultaneidade são permitidos, como leituras sujas ou leituras fantasma.

# Níveis de isolamento

- ▶ Um nível de isolamento mais baixo aumenta a capacidade de muitos usuários acessarem os dados ao mesmo tempo, porém aumenta o número de efeitos de simultaneidade que podem afetar os usuários.
- ▶ Um nível de isolamento mais alto reduz os tipos de efeitos de simultaneidade que os usuários podem encontrar, porém requer mais recursos do sistema e aumenta as chances de uma transação bloquear outra.
- ▶ Escolher o nível de isolamento apropriado depende de equilibrar os requisitos de integridade de dados do aplicativo em relação à sobrecarga de cada nível de isolamento.

# Fenômenos

- ▶ *Atualização Perdida*
  - ▶ *Leitura Suja*
  - ▶ *Leitura Não Repetível*
  - ▶ *Leitura Fantasma*
- 

# Atualização Perdida

(Escrita suja ou dirty write)

Duas transações que ocorrem simultaneamente atualizam o mesmo dado. Isto pode ocorrer em uma seqüência segundo a qual uma das atualizações é perdida.

$T_1$	$T_2$
read (X); X:=X-N;	read (X); X := X*0,9;
write (X); read (Y);	write (X);
Y:=Y+N; write (Y);	

o item X tem um valor incorreto porque a sua atualização feita por  $T_1$  foi perdida

# Leitura Suja

(Dirty Read ou dependência sem commit)

Leitura de dados não confirmados de uma linha existente, podendo ocasionar a leitura de uma informação nunca confirmada

T1 falha e tem que voltar o valor de X ao seu valor anterior. A esta altura, T2 já leu o valor temporário incorreto de X

T <sub>1</sub>	T <sub>2</sub>
read (X);	
X := X - N;	
write (X);	
	read (X);
	X := X + M;
	write (X);
read (Y);	
<b>(T<sub>1</sub> falha)</b>	

# Leitura Não Repetida

(Non repeatable read ou Análise inconsistente)

Duas leituras de dados na mesma transação não se repetem. Na segunda leitura, dados não existem ou foram modificados.

Quantidade de linhas NÃO muda!!!

<u>CPF</u>	Nome	Idade	Salario
1111111	Rafael	30	4.000
2222222	Lúcia	28	5.000
3333333	Ana	47	7.000

Leitura 1

<u>CPF</u>	Nome	Idade	Salario
1111111	Rafael	30	6.000
2222222	Lúcia	28	5.000
3333333	Ana	47	7.000

Leitura 2

# Leitura Fantasma

(Phantom read)

Na releitura de um conjunto de dados, surgem novas informações no conjunto.

Quantidade de linhas muda!!!

<u>CPF</u>	Nome	Idade	Salario
1111111	Rafael	30	4.000
2222222	Lúcia	28	5.000
3333333	Ana	47	7.000

Leitura 1

<u>CPF</u>	Nome	Idade	Salario
1111111	Rafael	30	4.000
2222222	Lúcia	28	5.000
3333333	Ana	47	7.000
4444444	Felipe	25	4.500

Leitura 2

# Níveis de Isolamento das Transações

- ▶ **READ UNCOMMITTED**
  - ▶ **READ COMMITTED**
  - ▶ **REPEATABLE READ**
  - ▶ **SERIALIZABLE**
- 
- ▶ **Exemplos:**
    - `START TRANSACTION SERIALIZABLE`
    - `SET TRANSACTION LEVEL SERIALIZABLE`
- 

# Níveis de Isolamento

NÍVEL DE ISOLAMENTO	PROBLEMA			
	Perda de Atualização	Leitura Suja	Leitura Não-repetida	Leitura Fantasma
READ UNCOMMITTED	Não permite	<b>Permite</b>	<b>Permite</b>	<b>Permite</b>
READ COMMITTED	Não permite	Não permite	<b>Permite</b>	<b>Permite</b>
REPEATABLE READ	Não permite	Não permite	Não permite	<b>Permite</b>
SERIALIZABLE	Não permite	Não permite	Não permite	Não permite

# Níveis de Isolamento

NÍVEL DE ISOLAMENTO	Leitura Suja
READ UNCOMMITTED	<b>Permite</b>
READ COMMITTED	Não permite
REPEATABLE READ	Não permite
SERIALIZABLE	Não permite

# Níveis de Isolamento

NÍVEL DE ISOLAMENTO	PROBLEMA			
	Perda de Atualização	Leitura Suja	Leitura Não-repetida	Leitura Fantasma
READ UNCOMMITTED	Não permite	<b>Permite</b>	<b>Permite</b>	<b>Permite</b>
READ COMMITTED	Não permite	Não permite	<b>Permite</b>	<b>Permite</b>
REPEATABLE READ	Não permite	Não permite	Não permite	<b>Permite</b>
SERIALIZABLE	Não permite	Não permite	Não permite	Não permite

**Q11** – O padrão SQL define diversos níveis de isolamento de transações. Dentre os níveis permitidos, encontra-se o nível de Leitura Confirmada (read committed) que é implementado atualmente por vários Sistemas Gerenciadores de Bancos de Dados (SGBDs). Em um SGBD operando em tal nível de isolamento, tentam-se executar duas transações (T1 e T2). Observe na tabela abaixo o registro do escalonamento previsto para os comandos dessas transações, em que A representa um determinado registro.

Considerando os dados registrados, conclui-se que

T1	Tempo	T2
Begin Transaction	t1	
Read(A)	t2	
...	t3	
...	t4	Begin Transaction
...	t5	Read(A)
...	t6	$A = A + 30$
...	t7	Write(A)
...	t8	Commit
Read(A)	t9	
$A = A - 50$	t10	
Write(A)	t11	
Rollback	t12	

- (A) a transação T1 não poderá executar o comando Rollback, pois a transação T2 executou o comando Commit.
- (B) a transação T1 terá dois resultados distintos (nos tempos t2 e t9) para o mesmo comando de consulta ao registro A.
- (C) o valor final do registro A, após a execução de ambas as transações T1 e T2, será igual ao valor inicial reduzido de 20.
- (D) o registro A terá o mesmo valor que tinha antes do início de ambas as transações, devido ao comando Rollback executado por T1, ao final da execução das transações T1 e T2.
- (E) esse escalonamento não pode ocorrer, pois o nível de isolamento utilizado impede a execução de duas transações em simultâneo, sendo que a transação T2 somente será executada pelo SGBD após o término da transação T1.

T1	Tempo	T2
Begin Transaction	t1	
Read(A)	t2	
...	t3	
...	t4	Begin Transaction
...	t5	Read(A)
...	t6	$A = A + 30$
...	t7	Write(A)
...	t8	Commit
Read(A)	t9	
$A = A - 50$	t10	
Write(A)	t11	
Rollback	t12	

(A) a transação T1 não poderá executar o comando Rollback, pois a transação T2 executou o comando Commit.

(B) a transação T1 terá dois resultados distintos (nos tempos t2 e t9) para o mesmo comando de consulta ao registro A.

(C) o valor final do registro A, após a execução de ambas as transações T1 e T2, será igual ao valor inicial reduzido de 20.

(D) o registro A terá o mesmo valor que tinha antes do início de ambas as transações, devido ao comando Rollback executado por T1, ao final da execução das transações T1 e T2.

(E) esse escalonamento não pode ocorrer, pois o nível de isolamento utilizado impede a execução de duas transações em simultâneo, sendo que a transação T2 somente será executada pelo SGBD após o término da transação T1.

## Q12 – CESGRANRIO – BNDES – INFRA – 2008

Considere o ambiente de uma aplicação multiusuário que acessa um sistema de gerência de bancos de dados relacional. Os usuários acessam a aplicação em simultâneo, submetendo transações concorrentes ao banco de dados. Todas as transações realizadas na base de dados, pela referida aplicação, estão definidas com o nível de isolamento READ COMMITTED (*leitura com efetivação*). O usuário João está executando, através da aplicação, uma transação T1, composta por vários comandos SQL. Neste caso, é correto afirmar que

- (A) o nível de isolamento adotado não irá impedir o aparecimento de linhas fantasmas (*phantom*) durante a execução de T1.
- (B) as atualizações de dados realizadas por João dentro de T1 podem ser lidas por outros usuários imediatamente, mesmo antes de João efetivar sua transação.
- (C) se João abortar a execução de T1 após ter executado, como parte da transação, comandos de atualização de dados, as referidas atualizações não poderão ser desfeitas.
- (D) no ambiente descrito, a execução intercalada de qualquer conjunto de transações será serializável.
- (E) devido à utilização do nível de isolamento especificado, enquanto João executar T1, nenhum outro usuário poderá executar comandos no banco de dados.

## Q12 – CESGRANRIO – BNDES – INFRA – 2008

Considere o ambiente de uma aplicação multiusuário que acessa um sistema de gerência de bancos de dados relacional. Os usuários acessam a aplicação em simultâneo, submetendo transações concorrentes ao banco de dados. Todas as transações realizadas na base de dados, pela referida aplicação, estão definidas com o nível de isolamento READ COMMITTED (*leitura com efetivação*). O usuário João está executando, através da aplicação, uma transação T1, composta por vários comandos SQL. Neste caso, é correto afirmar que

-  (A) o nível de isolamento adotado não irá impedir o aparecimento de linhas fantasmas (*phantom*) durante a execução de T1.
- (B) as atualizações de dados realizadas por João dentro de T1 podem ser lidas por outros usuários imediatamente, mesmo antes de João efetivar sua transação.
- (C) se João abortar a execução de T1 após ter executado, como parte da transação, comandos de atualização de dados, as referidas atualizações não poderão ser desfeitas.
- (D) no ambiente descrito, a execução intercalada de qualquer conjunto de transações será serializável.
- (E) devido à utilização do nível de isolamento especificado, enquanto João executar T1, nenhum outro usuário poderá executar comandos no banco de dados.

## Q13 – CESGRANRIO – BR DISTRIBUIDORA – INFRA – 2012

Sistemas Gerenciadores de Banco de Dados são concorrentes quando permitem mais de uma transação acessando o Banco de Dados (BD) ao mesmo tempo. Considere duas transações A e B, acessando o mesmo Banco de Dados. A transação B atualizou uma tupla no tempo  $t_1$ . No instante seguinte,  $t_2$ , a transação A leu essa mesma tupla. No instante  $t_3$ , a transação B foi cancelada, e ocorreu um ROLLBACK.

Esse problema de concorrência é denominado

- a) atualização perdida
- b) dependência sem commit
- c) análise inconsistente
- d) intenção de bloqueio
- e) nível de isolamento

## Q13 – CESGRANRIO – BR DISTRIBUIDORA – INFRA – 2012

Sistemas Gerenciadores de Banco de Dados são concorrentes quando permitem mais de uma transação acessando o Banco de Dados (BD) ao mesmo tempo. Considere duas transações A e B, acessando o mesmo Banco de Dados. A transação B atualizou uma tupla no tempo  $t_1$ . No instante seguinte,  $t_2$ , a transação A leu essa mesma tupla. No instante  $t_3$ , a transação B foi cancelada, e ocorreu um ROLLBACK.

Esse problema de concorrência é denominado

- a) atualização perdida
-  b) dependência sem commit
- c) análise inconsistente
- d) intenção de bloqueio
- e) nível de isolamento

# Equivalência de Escalonamentos

- ▶ Um escalonamento NÃO serial é seriável se ele for equivalente a um escalonamento serial
- ▶ Se um escalonamento é seriável ele é correto.

# Equivalência de Escalonamentos

## ▶ Equivalência por Resultado

- Produzem o mesmo estado final no BD
- Os resultados podem ser acidentalmente iguais

## ▶ Equivalência por Visão

- cada read lê o resultado de um mesmo write nos dois escalonamentos
- Se  $w_2(X)$  antecede  $r_1(X)$  em  $S_1$  isso tb ocorre em  $S_2$
- Se  $w_k(X)$  é a última operação em  $S_1$  isso tb ocorre em  $S_2$

## ▶ Equivalência por Conflito

- A ordem de qualquer duas operações conflitantes é a mesma nos dois escalonamentos

# Conflito

- ▶ Existe um conflito entre duas operações do escalonamento se as operações :
  - Pertencem a duas transações diferentes
  - Acessam o mesmo dado
  - Uma das operações é de escrita

$R_1(x); W_2(x);$

$R_2(x); W_1(x);$

$W_2(y); R_1(y);$



Estão em conflito

$R_1(x); W_2(y);$

$R_2(x); W_1(y);$

$W_2(y); R_1(x);$



Não estão em conflito

# Equivalência por conflito

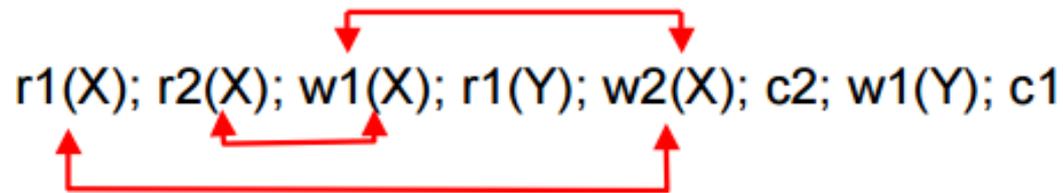
- ▶ Considere o escalonamento S das transações:

T1: r1(X); w1(X); r1(Y); w1(Y); c1

T2: r2(X); w2(X); c2;

S: r1(X); r2(X); w1(X); r1(Y); w2(X); c2; w1(Y); c1

Teremos 3 conflitos



T1	T2
r1(X)	
	r2(X)
w1(X)	
r1(Y)	
	w2(X)
	c2
w1(Y)	
c1	

# Equivalência por Conflito

- ▶ Dois escalonamentos são ditos equivalentes por conflito se a ordem de quaisquer duas operações conflitantes é a mesma nos dois escalonamentos
  - ▶ Se um escalonamento não serial é equivalente por conflito a um escalonamento serial, então ele é correto.
- 

# Equivalência por conflito

T1	T2
r1(X)	
w1(X)	
r1(Y)	
w1(Y)	
	r2(X)
	w2(X)
	r2(Y)
	w2(Y)

Escalonamento serial:

S1: r1(X); w1(X); r1(Y); w1(Y); r2(X); w2(X); r2(Y); w2(Y);

► Conflitos:

- r1(X) com w2(X); w1(X) com r2(X); w1(X) com w2(X);
- r1(Y) com w2(Y); w1(Y) com r2(Y); w1(Y) com w2(Y);

# Equivalência por conflito

T1	T2
r1(X)	
w1(X)	
	r2(X)
	w2(X)
r1(Y)	
w1(Y)	
	r2(Y)
	w2(Y)

- S1 e S3 são equivalentes por conflito.
- S3 é seriável pois é equivalente por conflito a um escalonamento serial, então ele é correto.

Escalonamento concorrente:

S3: r1(X); w1(X); r2(X); w2(X); r1(Y); w1(Y); r2(Y); w2(Y);

► Conflitos:

- r1(X) com w2(X); w1(X) com r2(X); w1(X) com w2(X);
- r1(Y) com w2(Y); w1(Y) com r2(Y); w1(Y) com w2(Y);

# Equivalência por conflito

T1	T2
r1(X)	
	r2(X)
w1(X)	
	w2(X)
r1(Y)	
w1(Y)	
	r2(Y)
	w2(Y)

- S1 e S4 **NÃO** são equivalentes por conflito.
- Ordem das operações conflitantes é diferente.

Escalonamento concorrente:

S4: r1(X); r2(X); w1(X); w2(X); r1(Y); w1(Y); r2(Y); w2(Y);

## ▶ Conflitos:

- r1(X) com w2(X); **r2(X) com w1(X)**; w1(X) com w2(X);
- r1(Y) com w2(Y); w1(Y) com r2(Y); w1(Y) com w2(Y);

## Testando a Serialização por Conflito de um Plano

- ▶ Há um algoritmo para determinar se um plano é serializável de conflito ou não.
- ▶ O algoritmo verifica as operações ler\_item e escrever\_item de um plano e constrói um grafo de precedência (grafo de serialização).
- ▶ O grafo de serialização é um grafo orientado  $G(N,E)$ , onde  $N$  é o conjunto de transações do plano.

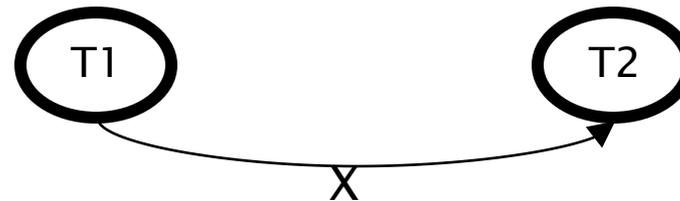
# Testando a Serialização por Conflito de um Plano

- ▶ Algoritmo para testar a serialização de conflito de um plano S
  1. Para cada transação  $T_i$  participante do plano S, criar um nó  $T_i$  no grafo.
  2. Para cada caso em S em que  $T_j$  executar um `ler_item(Z)` depois que uma  $T_i$  executar um `escrever_item(Z)`, criar um arco ( $T_i \rightarrow T_j$ ) no grafo.
  3. Para cada caso em S em que  $T_j$  executar um `escrever_item(Z)` depois que uma  $T_i$  executar um `ler_item(Z)`, criar um arco ( $T_i \rightarrow T_j$ ) no grafo.
  4. Para cada caso em S em que  $T_j$  executar um `escrever_item(Z)` depois que uma  $T_i$  executar um `escrever_item(Z)`, criar um arco ( $T_i \rightarrow T_j$ ) no grafo.
  5. O plano S será serializável se, e apenas se, o grafo não contiver ciclos.

# Testando a Serialização por Conflito de um Plano

## ▶ Exemplos:

T1	T2
r1(X)	
$X = X - N$	
w1(X)	
r1(Y)	
$Y = Y + N$	
w1(Y)	
	r2(X)
	$X = X + M$
	w2(X)

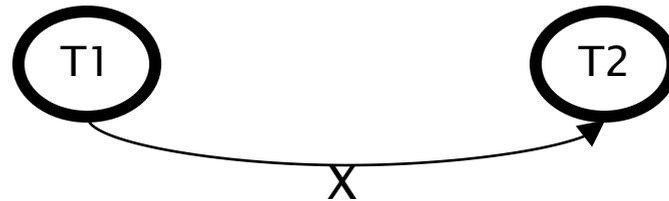


Plano é serializável pois o grafo não contém ciclos

# Testando a Serialização por Conflito de um Plano

## ▶ Exemplos:

T1	T2
r1(X)	
$X = X - N$	
w1(X)	
	r2(X)
	$X = X + M$
	w2(X)
r1(Y)	
$Y = Y + N$	
w1(Y)	

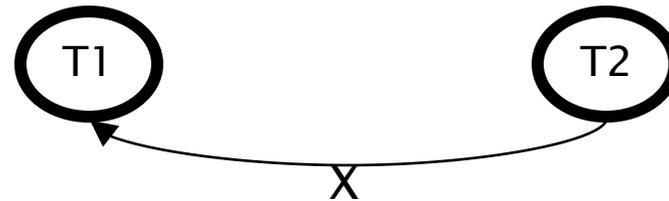


Plano é serializável pois o grafo não contém ciclos

# Testando a Serialização por Conflito de um Plano

## ▶ Exemplos:

T1	T2
	r2(X)
	X = X + M
	w2(X)
r1(X)	
X = X - N	
w1(X)	
r1(Y)	
Y = Y + N	
w1(Y)	

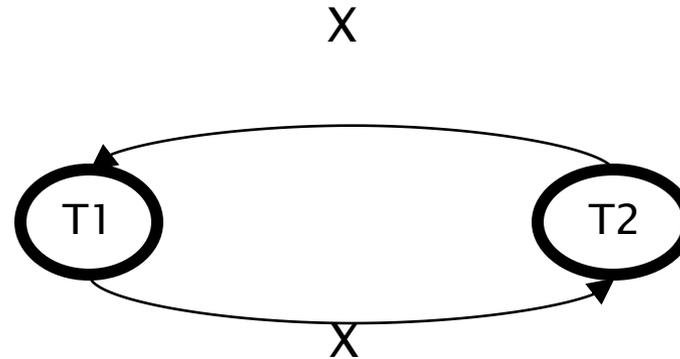


Plano é serializável pois o grafo não contém ciclos

# Testando a Serialização por Conflito de um Plano

## Exemplos:

T1	T2
r1(X)	
X = X - N	
	r2(X)
	X = X + M
w1(X)	
r1(Y)	
	w2(X)
Y = Y + N	
w1(Y)	



Plano NÃO é serializável pois o grafo contém ciclos

# Equivalência de Visão

- ▶ A definição de equivalência de planos chamada de equivalência de visão é menos restritiva do que a equivalência de conflito.
- ▶ Dois planos  $S$  e  $S'$  são visão equivalentes se as três condições seguintes forem atendidas:
  1. O conjunto de transações participantes em  $S$  e  $S'$  seja o mesmo.
  2. Para toda operação  $ri(X)$  de  $T_i$  em  $S$ , se o valor  $X$  lido pela operação tiver sido produzido por uma operação  $wj(X)$  de  $T_j$  (ou se for o valor original de  $X$  antes do plano iniciar), a mesma condição deverá ser garantida para o valor 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  $T_k$  deverá ser a última operação a gravar  $Y$  em  $S'$ .

# Equivalência de Visão

## ▶ Idéia básica

- enquanto cada  $read(X)$  de uma  $T_x$  ler o resultado de uma mesmo  $write(X)$  em  $E'$  e  $E$ , em ambos os escalonamentos,  $T_x$  tem a mesma visão do resultado
- se o último  $write(X)$  é feito pela mesma transação em  $E'$  e  $E$ , então o estado final do BD será o mesmo em ambos os escalonamentos

## ▶ Exemplo

$H_{\text{serial}} = r1(X) \text{ w1(X) c1 w2(X) c2 w3(X) c3}$

$H_{\text{exemplo}} = r1(X) \text{ w2(X) w1(X) w3(X) c1 c2 c3}$

- $H_{\text{exemplo}}$  **não é serializável em conflito**, mas é serializável em visão

# Equivalência de Visão

- ▶ A serializabilidade de visão é **menos restritiva** que a serializabilidade em conflito.
  - um escalonamento  $E'$  serializável em conflito também é serializável em visão, porém o contrário nem sempre é verdadeiro.
- ▶ A serializabilidade de visão é **muito mais complexa de verificar** que a serializabilidade em conflito.

## Q14 – CESGRANRIO – TRANSPETRO – INFRA – 2012

Ao gerenciador de transações de um Sistema de Gerência de Banco de Dados é apresentado o escalonamento a seguir.

$r1(X); r2(X); w1(X); r1(Y); w2(X); w1(Y);$

Nesse escalonamento, considere que  $r1$  e  $w1$  correspondem a operações de leitura e de escrita da transação 1, respectivamente.

Considere, também, que  $r2$  e  $w2$  correspondem a operações de leitura e de escrita da transação 2, respectivamente.

Por fim, considere que as operações de leitura e escrita são seguidas pelos itens de dados ( $X$  ou  $Y$ ) apresentados entre parênteses.

No escalonamento fornecido, duas operações que **NÃO** estão em conflito são

- (A)  $r1(X)$  e  $w2(X)$
- (B)  $r1(Y)$  e  $w2(Y)$
- (C)  $r1(X)$  e  $w1(X)$
- (D)  $r2(X)$  e  $w1(X)$
- (E)  $w1(X)$  e  $w2(X)$

## Q14 – CESGRANRIO – TRANSPETRO – INFRA – 2012

Ao gerenciador de transações de um Sistema de Gerência de Banco de Dados é apresentado o escalonamento a seguir.

$r1(X); r2(X); w1(X); r1(Y); w2(X); w1(Y);$

Nesse escalonamento, considere que  $r1$  e  $w1$  correspondem a operações de leitura e de escrita da transação 1, respectivamente.

Considere, também, que  $r2$  e  $w2$  correspondem a operações de leitura e de escrita da transação 2, respectivamente.

Por fim, considere que as operações de leitura e escrita são seguidas pelos itens de dados ( $X$  ou  $Y$ ) apresentados entre parênteses.

No escalonamento fornecido, duas operações que **NÃO** estão em conflito são

(A)  $r1(X)$  e  $w2(X)$

(B)  $r1(Y)$  e  $w2(Y)$

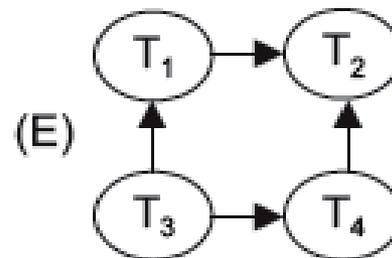
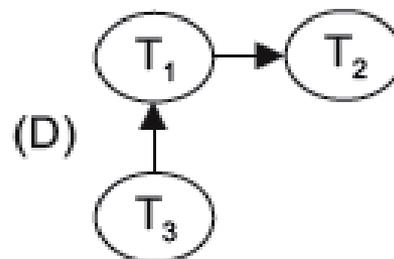
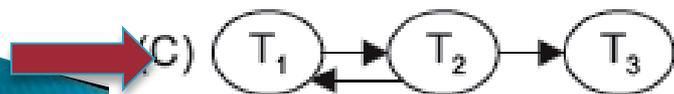
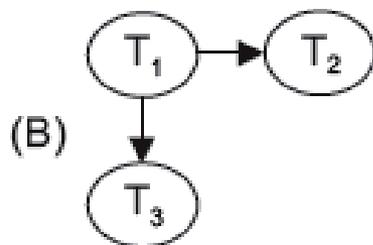
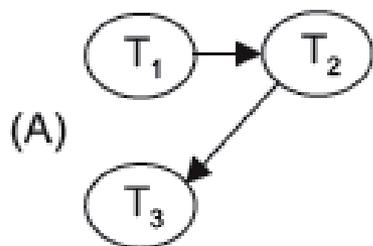
 (C)  $r1(X)$  e  $w1(X)$

(D)  $r2(X)$  e  $w1(X)$

(E)  $w1(X)$  e  $w2(X)$

# Q15 – CESGRANRIO – BNDES – SUPORTE – 2008

No âmbito de banco de dados, que grafo de precedência representa um escalonamento que **NÃO** é serializável quanto ao conflito?



# Gabarito

Q1 - C

Q2 - E

Q3 - E

Q4 - E

Q5 - A

Q6 - B

Q7 - E

Q8 - D

Q9 - D

Q10 - C

Q11 - B

Q12 - A

Q13 - B

Q14 - C

Q15 - C

# BANCO DE DADOS

# Transações

Prof. Eduardo Neves  
[edumneves@gmail.com](mailto:edumneves@gmail.com)

Dúvidas de Banco de Dados:  
Fórum do curso ☺

Artigos do Blog:  
<http://goo.gl/Sm6JgV>

# Técnicas de Recuperação

»» BANCO DE DADOS

# Introdução

- ▶ A recuperação de falhas existe para garantir as propriedades de atomicidade e durabilidade de transações.
- ▶ O sistema de recuperação(restauração) de falhas é responsável pela restauração do banco de dados para um estado – o que havia antes da ocorrência de uma falha.

# Tipos de Falhas

## ▶ Lógicas

- Overflow
- Divisão por zero
- Chamada equivocada de parâmetros
- Exceções não tratadas devidamente
- Solicitação do usuário

## ▶ Físicas

- Incêndio
- Desastres
- Queima do ar condicionado
- Perda de buffer
- Erros de hardware e software

# Log

- ▶ O LOG é uma seqüência de registros de log que mantém um arquivo atualizado sobre as atividades realizadas com os dados de um banco de dados.
- ▶ Sempre que uma operação de escrita é realizada é essencial que o registro de log equivalente seja criado antes que o banco de dados seja modificado;
- ▶ A inutilização de uma operação de escrita (undo) pode ser realizada com o uso do valor antigo;
- ▶ O log deve residir em armazenamento estável;
- ▶ O log contém o registro completo de todas as atividades realizadas com o banco de dados;
- ▶ O tamanho do log pode se tornar absurdamente grande.

# Registro de Log

- ▶ Um registro de atualização de log descreve uma única operação (de escrita) com um dado e possui os seguintes campos:
  - $T_i$ , o identificador da transação;
  - $X_j$ , o identificador do item de dado;  $\langle T_i, X_j, V1, V2 \rangle$
  - $V1$ , o valor antigo;
  - $V2$ , o valor novo;
- ▶ Outros registros de log:
  - Início de transação;  $\langle T_i, \text{start} \rangle$
  - Efetivação de transação;  $\langle T_i, \text{commit} \rangle$
  - Abortar a transação.  $\langle T_i, \text{abort} \rangle$
  - Ler dados  $\langle T_i, \text{read}, X \rangle$

Em uma situação de falha, o Sistema Gerenciador de Banco de Dados (SGBD) busca restaurar o banco de dados ao estado consistente mais recente antes do momento da falha. Para realizar essa função, o sistema precisa manter informações sobre as mudanças que foram aplicadas aos itens de dados pelas diversas transações. Tais mudanças são armazenadas no

- a) catálogo do banco de dados.
- b) arquivo de log do banco de dados.
- c) dicionário do banco de dados.
- d) arquivo de metadados do banco de dados.

Em uma situação de falha, o Sistema Gerenciador de Banco de Dados (SGBD) busca restaurar o banco de dados ao estado consistente mais recente antes do momento da falha. Para realizar essa função, o sistema precisa manter informações sobre as mudanças que foram aplicadas aos itens de dados pelas diversas transações. Tais mudanças são armazenadas no

- a) catálogo do banco de dados.
-  b) arquivo de log do banco de dados.
- c) dicionário do banco de dados.
- d) arquivo de metadados do banco de dados.

## Q17 – ESAF – MF – 2013

No acesso a dados baseado em log, um registro de log de atualização contém:

- a) Identificador de transposição. Identificador de item de dados. Valor desejado. Valor de referência.
- b) Identificador de transição. Identificador de item de dados. Valor desejado. Valor de referência.
- c) Identificador de transação. Identificador de item de dados. Valor antigo. Novo valor.
- d) Identificador de transposição. Identificador de item de dados. Valor codificado. Novo valor.
- e) Momento da transição. Identificador de chaves de acesso. Valor antigo. Novo valor.

## Q17 – ESAF – MF – 2013

No acesso a dados baseado em log, um registro de log de atualização contém:

- a) Identificador de transposição. Identificador de item de dados. Valor desejado. Valor de referência.
- b) Identificador de transição. Identificador de item de dados. Valor desejado. Valor de referência.
-  c) Identificador de transação. Identificador de item de dados. Valor antigo. Novo valor.
- d) Identificador de transposição. Identificador de item de dados. Valor codificado. Novo valor.
- e) Momento da transição. Identificador de chaves de acesso. Valor antigo. Novo valor.

# Q18 – FGV – DPE-MT – 2015

Um dos requisitos observados pelos sistemas gerenciadores de bancos de dados é a garantia da durabilidade de uma transação. Se um conjunto de operações tiver sido confirmado (committed), o sistema deve manter esse resultado no caso de recuperação de falhas, mesmo quando há perda de dados.

Assinale a opção que indica a característica principal da técnica comumente usada para a obtenção dessa garantia nas implementações.

- a) Algoritmos de bloqueio
- b) Arquivos de log
- c) Backup diferencial
- d) Replicação de banco de dados
- e) Serialização

# Q18 – FGV – DPE-MT – 2015

Um dos requisitos observados pelos sistemas gerenciadores de bancos de dados é a garantia da durabilidade de uma transação. Se um conjunto de operações tiver sido confirmado (committed), o sistema deve manter esse resultado no caso de recuperação de falhas, mesmo quando há perda de dados.

Assinale a opção que indica a característica principal da técnica comumente usada para a obtenção dessa garantia nas implementações.

- a) Algoritmos de bloqueio
-  b) Arquivos de log
- c) Backup diferencial
- d) Replicação de banco de dados
- e) Serialização

# Técnicas de Recuperação de Falhas

- ▶ Atualização Adiada ou atualização postergada
  - Algoritmo NO-UNDO/REDO
- ▶ Atualização Imediata
  - Algoritmo UNDO/REDO

# Atualização Adiada

- ▶ O BD não é atualizado até que a transação atinja o seu ponto de *commit*.
- ▶ Nessa técnica, somente o valor novo do item de dado que sofreu alteração é necessário ser guardado no log.
- ▶ As escritas são realizadas após a efetivação parcial da transação;
- ▶ No caso de falhas:
  - Antes das escritas no BD:
    - os registros no log serão ignorados (o valor antigo do item de dado permanecerá).
  - Após as escritas no BD:
    - executa-se operações de redo.

# Exemplo:

Valores iniciais:

A = 1000

B = 2000

C = 700

T0	T1	Log
Read(A);		<T0, start>
A = A - 50;		<T0, A, 950>
Write(A);		<T0, B, 2050>
Read(B);		<T0, commit>
B = B + 50;		<T1, start>
Write (B)		<T1, C, 600>
	Read(C);	<T1, commit>
	C = C - 100;	
	Write(C);	

# Recuperação

- ▶ Procedimento de recuperação em caso de falhas, que resultem em perda de informação no armazenamento volátil:
- ▶ Redo( $T_i$ ): Refazer  $T_i \rightarrow$  define o valor de todos os itens de dados atualizados pela transação  $T_i$  para os novos valores.
- ▶ Uma transação  $T_i$  deverá ser refeita, se e somente se, o sistema de recuperação encontrar os registros  $\langle T_i, \text{start} \rangle$  e  $\langle T_i, \text{commit} \rangle$  no log.

# Exemplos:

## Log A

---

<T0, start>  
<T0, A, 950>  
<T0, B, 2050>

Não tem commit.  
Nenhum redo é necessário.

## Log B

---

<T0, start>  
<T0, A, 950>  
<T0, B, 2050>  
<T0, commit>  
<T1, start>  
<T1, C, 600>

É necessário fazer redo (T0)  
Pois há um <T0, commit>

## Log C

---

<T0, start>  
<T0, A, 950>  
<T0, B, 2050>  
<T0, commit>  
<T1, start>  
<T1, C, 600>  
<T1, commit>

É necessário fazer redo (T0)  
Seguido de um redo (T1)  
Pois há um <T0, commit> e  
Depois um <T1, commit>

# Atualização Imediata

- ▶ Permite que as modificações no banco de dados sejam realizadas enquanto as transações ainda estão num estado ativo.
  - ▶ Neste esquema de modificação de BD, o log deverá armazenar o valor antigo e o valor novo oriundos das operações de write.
  - ▶ As informações podem ser atualizadas no banco de dados assim que o log esteja salvo em meio estável.
- 

# Exemplo:

Valores iniciais:

A = 1000

B = 2000

C = 700

T0	T1	Log	BD
Read(A);		<T0, start>	
A = A - 50;		<T0, A, 1000, 950>	A = 950
Write(A);		<T0, B, 2000, 2050>	B = 2050
Read(B);		<T0, commit>	
B = B + 50;		<T1, start>	
Write (B)		<T1, C, 700, 600>	C = 600
	Read(C);	<T1, commit>	
	C = C - 100;		
	Write(C);		

# Recuperação

- ▶ Undo( $T_i$ ): Desfazer  $T_i$ 
  - retorna aos valores antigos todos os itens de dados atualizados pela transação  $T_i$ .
- ▶ Redo( $T_i$ ): Refazer  $T_i$ 
  - ajusta os valores de todos os itens de dados atualizados pela transação  $T_i$  para os novos valores.

# Exemplos:

## Log A

---

<T0, start>  
<T0, A, 1000, 950>  
<T0, B, 2000, 2050>

Undo(T0):  
B é restaurado para 2000  
A é restaurado para 1000

## Log B

---

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

Undo(T1) e Redo (T0)  
C é restaurado para 700  
A retorna para 950  
B retorna para 2050

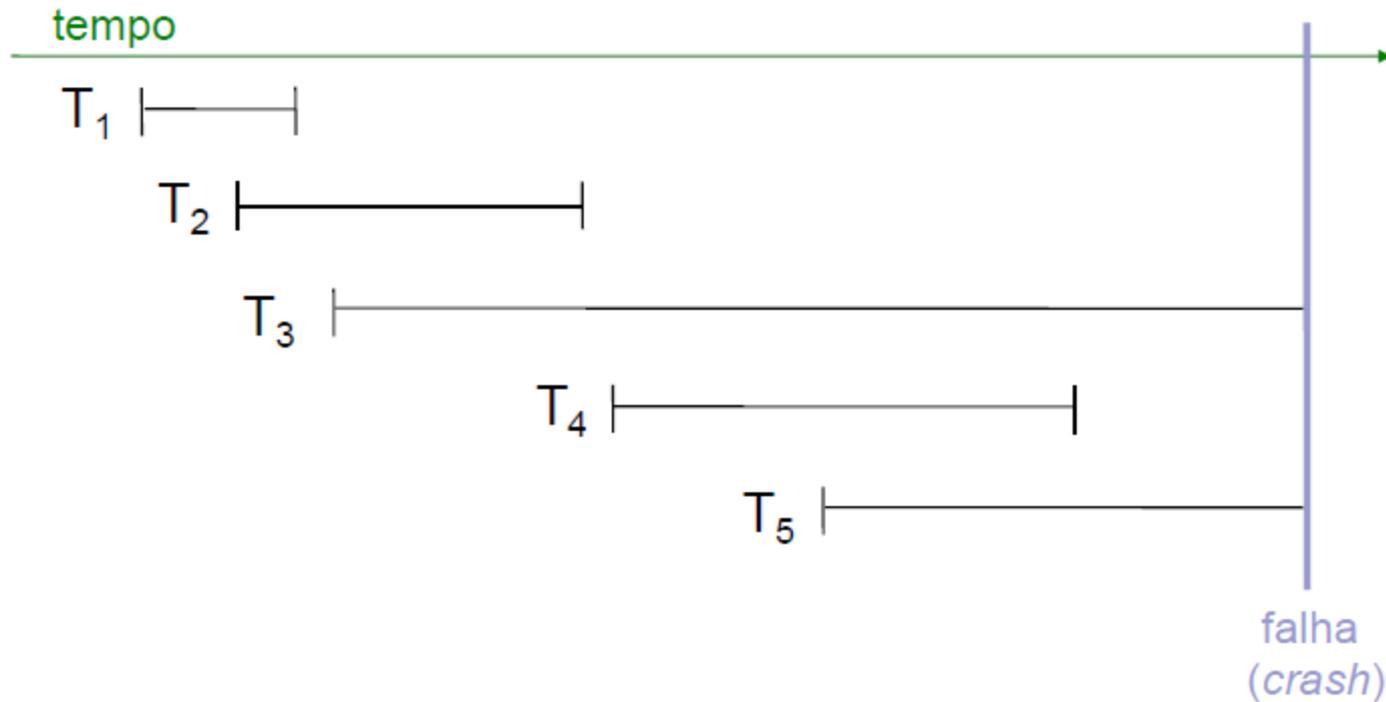
## Log C

---

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

Redo(T0) e redo (T1)  
A vale 950  
B vale 2050  
C vale 600

# Exemplos:



Undo:  $T_3$  e  $T_5$

Redo:  $T_1$ ,  $T_2$  e  $T_4$

# Checkpoints (pontos de controle)

- ▶ Na existência de uma falha, o sistema de recuperação deve, a princípio, percorrer todo o log para saber quais transações devem ser desfeitas.
- ▶ Dificuldades:
  - O processo de pesquisa consome tempo
  - Muitas das transações que necessitam ser refeitas já escreveram suas atualizações no BD.
    - Embora refazê-las não cause dano algum, a recuperação se tornará mais onerosa.

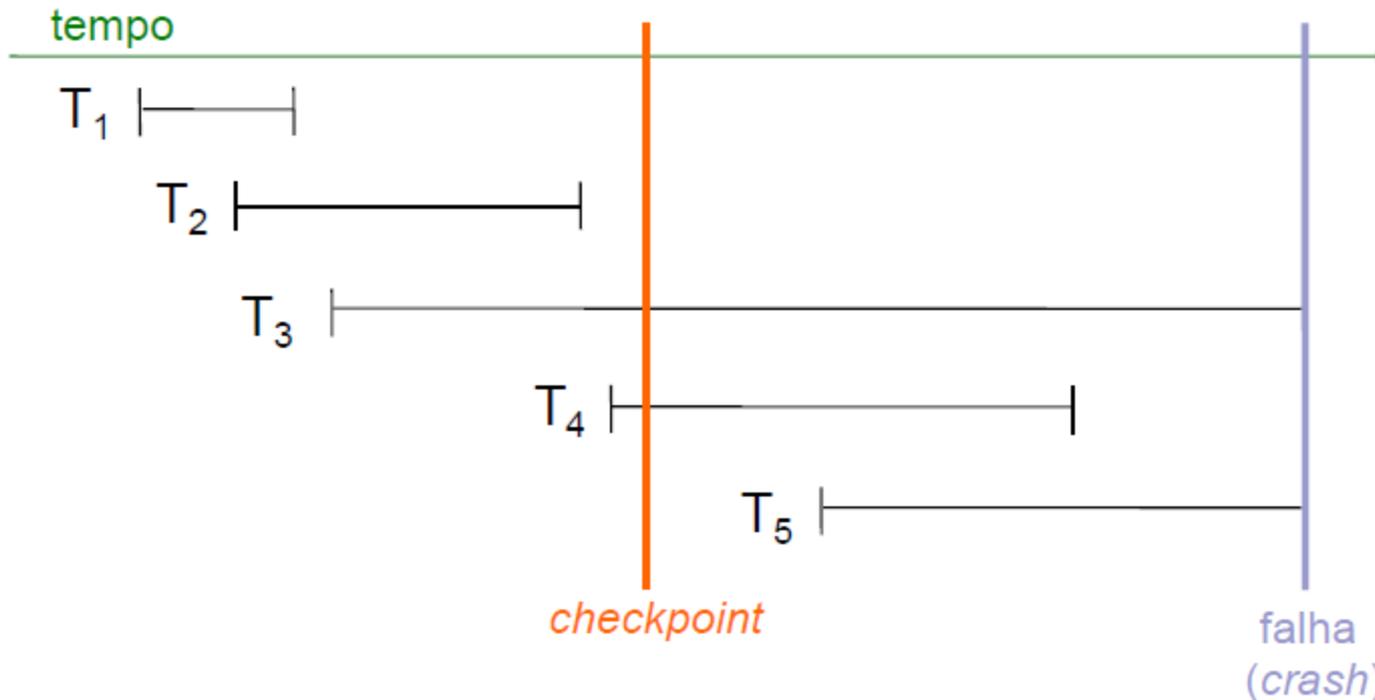
# Checkpoints (pontos de Controle)

- ▶ Solução: introdução de checkpoints (pontos de controle).
- ▶ Os checkpoints são registros inseridos no log periodicamente e exigem a execução da seqüência de operações abaixo:
  - Suspender, temporariamente, a execução de todas as transações.
  - Forçar as escritas das operações de write das transações, da memória principal para o disco.
  - Escrever o checkpoint no log e forçar a escrita no log em disco.
  - Reassumir a execução das transações

# Checkpoints

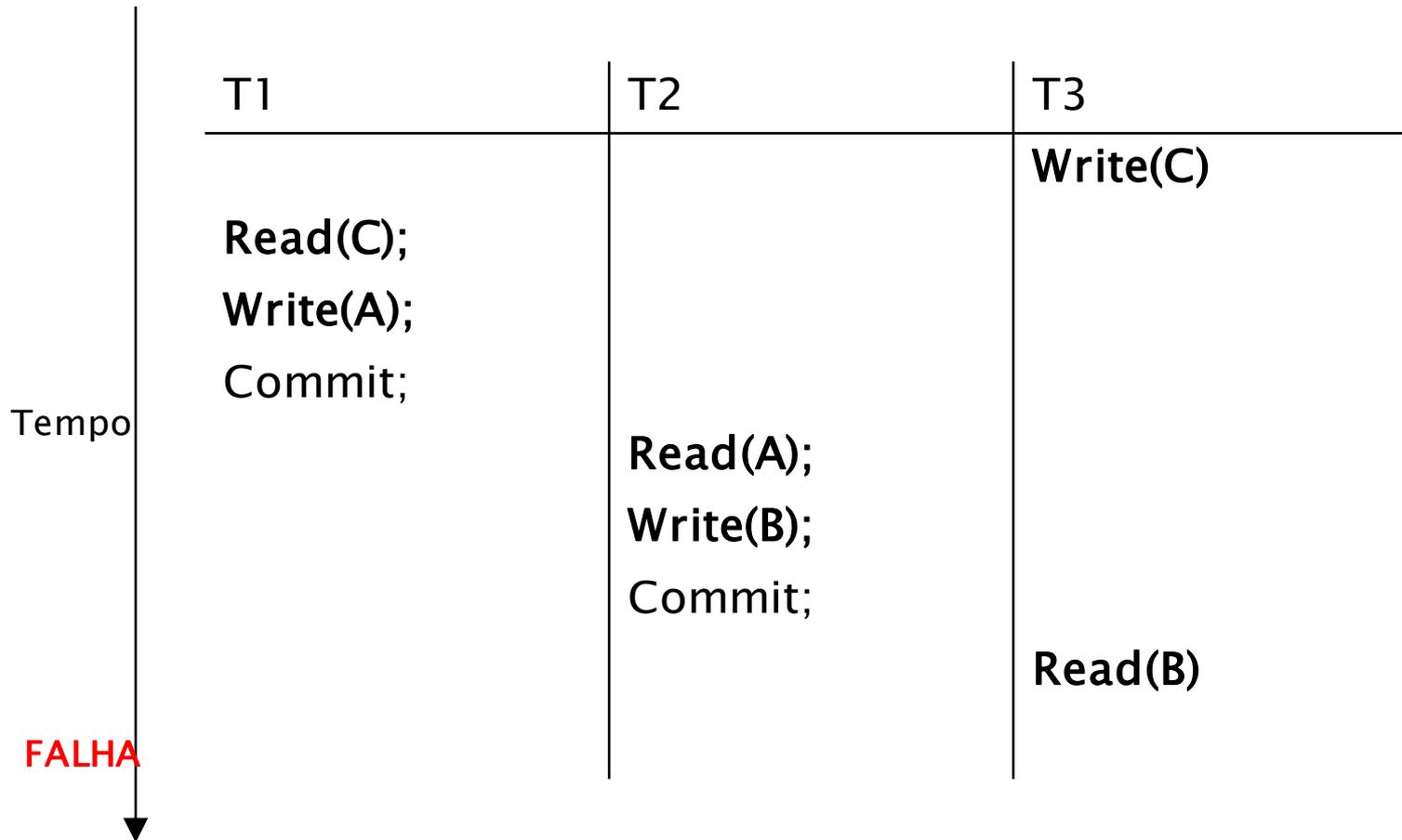
- ▶ Se  $T_i$  é efetivada antes do checkpoint, o registro  $\langle T_i, \text{commit} \rangle$  aparece no log antes do checkpoint, quaisquer modificações feitas por  $T_i$  ou já foram escritas no banco de dados antes do checkpoint ou o foram como parte do checkpoint propriamente dito.
- ▶ Assim, no momento de recuperação não haverá necessidade de uma operação de redo sobre  $T_i$ .

# UNDO/REDO com checkpoint



- ▶ UNDO: T3 e T5
- ▶ REDO: T4
- ▶ T1 e T2 concluíram e estão, garantidamente, no BD ⇒ não sofrem redo;
- ▶ T4 concluiu, mas suas atualizações não necessariamente estão no BD ⇒ sofre redo;
- ▶ T3 e T5 não concluíram ⇒ sofrem undo.

# Rollback em Cascata (Reversão 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

## Q19 – UEAP – CS–UFG – 2014

As operações refazer (REDO) e desfazer (UNDO) são usadas no processo de recuperação de banco de dados. Na recuperação baseada em atualização adiada, os dados atualizados por uma transação não podem ser gravados no banco de dados antes do commit da transação. O algoritmo para esse tipo de recuperação é denominado

- a) UNDO/REDO.
- b) UNDO/NO-REDO.
- c) NO-UNDO/NO-REDO.
- d) NO-UNDO/REDO.

## Q19 – UEAP – CS–UFG – 2014

As operações refazer (REDO) e desfazer (UNDO) são usadas no processo de recuperação de banco de dados. Na recuperação baseada em atualização adiada, os dados atualizados por uma transação não podem ser gravados no banco de dados antes do commit da transação. O algoritmo para esse tipo de recuperação é denominado

- a) UNDO/REDO.
- b) UNDO/NO-REDO.
- c) NO-UNDO/NO-REDO.
-  d) NO-UNDO/REDO.

O arquivo de log é usado no processo de recuperação de banco de dados. O registro de checkpoint é gravado periodicamente no log, no ponto em que o sistema grava, no banco de dados em disco, todos os blocos de memória do sistema gerenciador de banco de dados que foram modificados. Essa operação torna o processo de recuperação mais eficiente, porque

- a) aumenta o número de operações desfazer (UNDO).
- b) aumenta o número de operações refazer (REDO).
- c) reduz o número de operações desfazer (UNDO).
- d) reduz o número de operações refazer (REDO).

O arquivo de log é usado no processo de recuperação de banco de dados. O registro de checkpoint é gravado periodicamente no log, no ponto em que o sistema grava, no banco de dados em disco, todos os blocos de memória do sistema gerenciador de banco de dados que foram modificados. Essa operação torna o processo de recuperação mais eficiente, porque

- a) aumenta o número de operações desfazer (UNDO).
- b) aumenta o número de operações refazer (REDO).
- c) reduz o número de operações desfazer (UNDO).
- d) reduz o número de operações refazer (REDO).

## Q21 – FCC – TJUAP – 2014

Uma propriedade importante quando se considera a recuperação

de um banco de dados é constituída pelos chamados pontos de verificação (checkpoints). Um checkpoint consiste

de um registro, cuja função é indicar que

- (A) o banco de dados sofreu uma paralisação programa da para manutenção.
- (B) houve uma falha no conjunto de discos utilizado para armazenamento dos dados.
- (C) houve uma falha de comunicação entre as máquinas servidora e cliente
- (D) o banco de dados passou por uma operação de backup completo.
- (E) todas as transações encerradas anteriormente ao checkpoint já foram confirmadas.

## Q21 – FCC – TJUAP – 2014

Uma propriedade importante quando se considera a recuperação

de um banco de dados é constituída pelos chamados pontos de verificação (checkpoints). Um checkpoint consiste

de um registro, cuja função é indicar que

- (A) o banco de dados sofreu uma paralisação programa da para manutenção.
- (B) houve uma falha no conjunto de discos utilizado para armazenamento dos dados.
- (C) houve uma falha de comunicação entre as máquinas servidora e cliente
- (D) o banco de dados passou por uma operação de backup completo.
- (E) todas as transações encerradas anteriormente ao checkpoint já foram confirmadas.

## Q22 – FUMARC – TJMG – 2012

\_\_\_\_\_ é um tipo de entrada no log de operações das transações de um banco de dados escrita quando o SGBD grava no banco de dados os buffers de memória que tiverem sido modificados. Esse tipo de entrada garante que buffers modificados sejam gravados em disco mesmo que suas transações não tiverem sido efetivadas. Complete a lacuna com a opção correta dentre as opções seguintes.

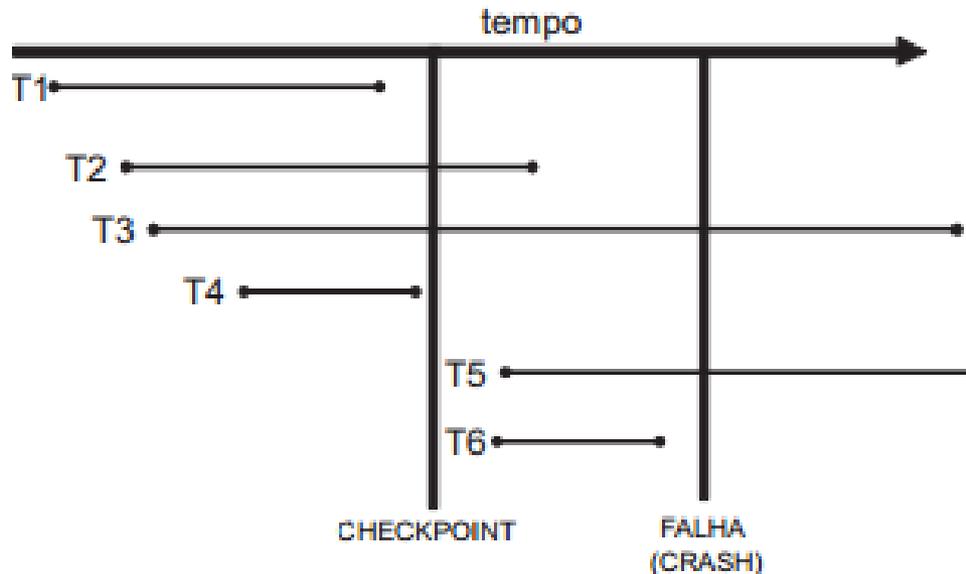
- a) Rollback
- b) Write
- c) Commit
- d) Checkpoint

## Q22 – FUMARC – TJMG – 2012

\_\_\_\_\_ é um tipo de entrada no log de operações das transações de um banco de dados escrita quando o SGBD grava no banco de dados os buffers de memória que tiverem sido modificados. Esse tipo de entrada garante que buffers modificados sejam gravados em disco mesmo que suas transações não tiverem sido efetivadas. Complete a lacuna com a opção correta dentre as opções seguintes.

- a) Rollback
- b) Write
- c) Commit
-  d) Checkpoint

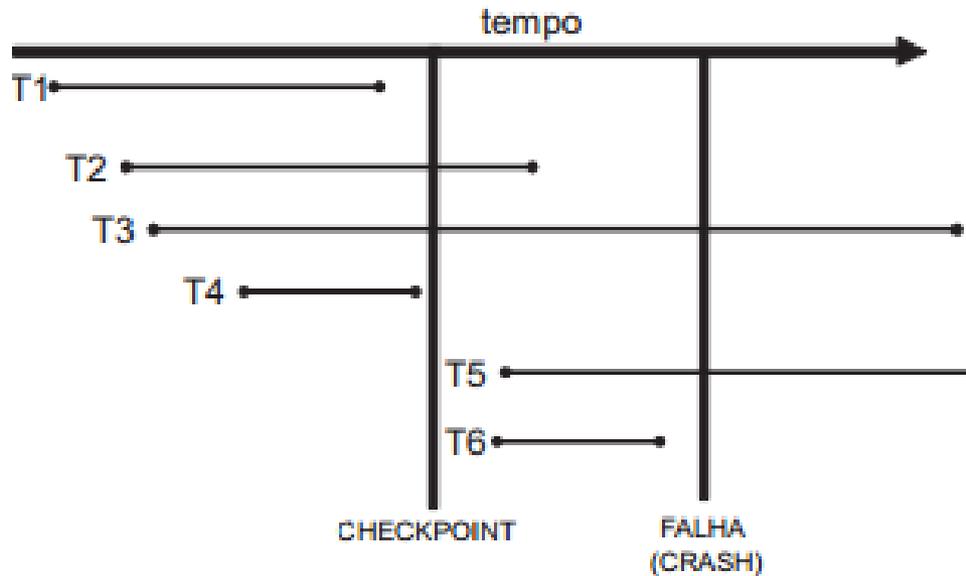
## Q23 – CESGRANRIO – Liquigas – Adm. Dados – 2012



Considerando as técnicas e os procedimentos de recuperação em caso de falhas, se a técnica de recuperação usada for a de atualização imediata, o que acontece com as transações T1, T2, T3, T4, T5 e T6, que estão sendo executadas em um Sistema Gerenciador de Banco de Dados, conforme apresentado na figura, após a falha?

- (A) T1, T2, T3, T4, T5 e T6 são desfeitas.
- (B) T1, T2 e T4 são refeitas, T3 e T5 são ignoradas, e T6 é desfeita.
- (C) T1, T2 e T4 estão concluídas, T3 e T5 são refeitas, e T6 é ignorada.
- (D) T1 e T4 estão concluídas, T2 e T6 são refeitas, e T3 e T5 são desfeitas.
- (E) T1 e T4 são desfeitas, T2 e T6 estão concluídas, e T3 e T5 são refeitas.

## Q23 – CESGRANRIO – Liquigas – Adm. Dados – 2012



Considerando as técnicas e os procedimentos de recuperação em caso de falhas, se a técnica de recuperação usada for a de atualização imediata, o que acontece com as transações T1, T2, T3, T4, T5 e T6, que estão sendo executadas em um Sistema Gerenciador de Banco de Dados, conforme apresentado na figura, após a falha?

- (A) T1, T2, T3, T4, T5 e T6 são desfeitas.
- (B) T1, T2 e T4 são refeitas, T3 e T5 são ignoradas, e T6 é desfeita.
- (C) T1, T2 e T4 estão concluídas, T3 e T5 são refeitas, e T6 é ignorada.
- (D) T1 e T4 estão concluídas, T2 e T6 são refeitas, e T3 e T5 são desfeitas.
- (E) T1 e T4 são desfeitas, T2 e T6 estão concluídas, e T3 e T5 são refeitas.

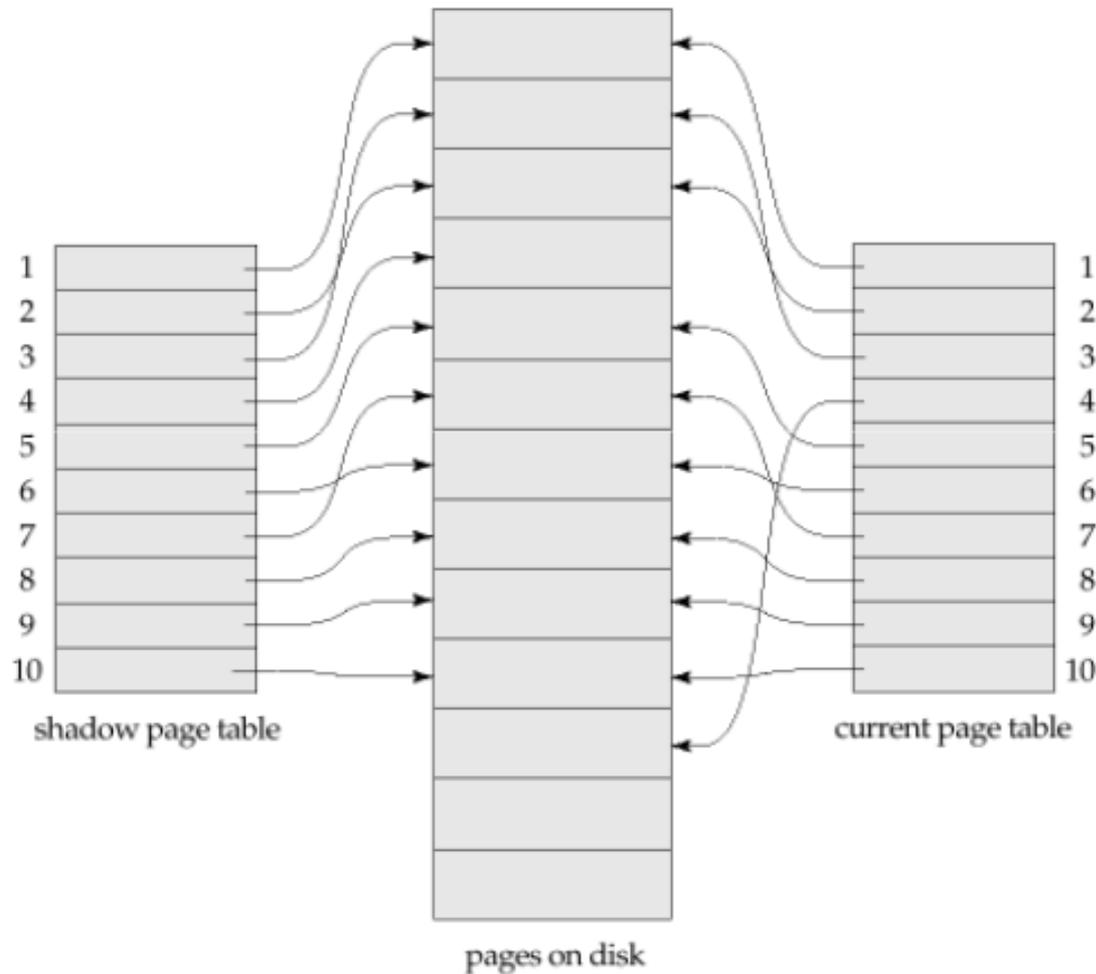
# Paginação Sombra (Shadow Paging)

- ▶ Shadow Paging é uma alternativa à recuperação baseada em log;
- ▶ Idéia: Manter duas tabelas de páginas durante o tempo de vida de uma transação – a Tabela de Páginas Corrente (TPC), e a Tabela de Páginas Shadow (TPS);
- ▶ Armazena a Tabela de Páginas Shadow em meio não-volátil, tal que o estado do BD anterior à execução da transação possa ser recuperado.
- ▶ Ao iniciar, ambas as tabelas de páginas são indênticas.
- ▶ Somente a tabela de páginas correntes é usada para cada item de dados acessados durante a execução da transação.

# Paginação Shadow

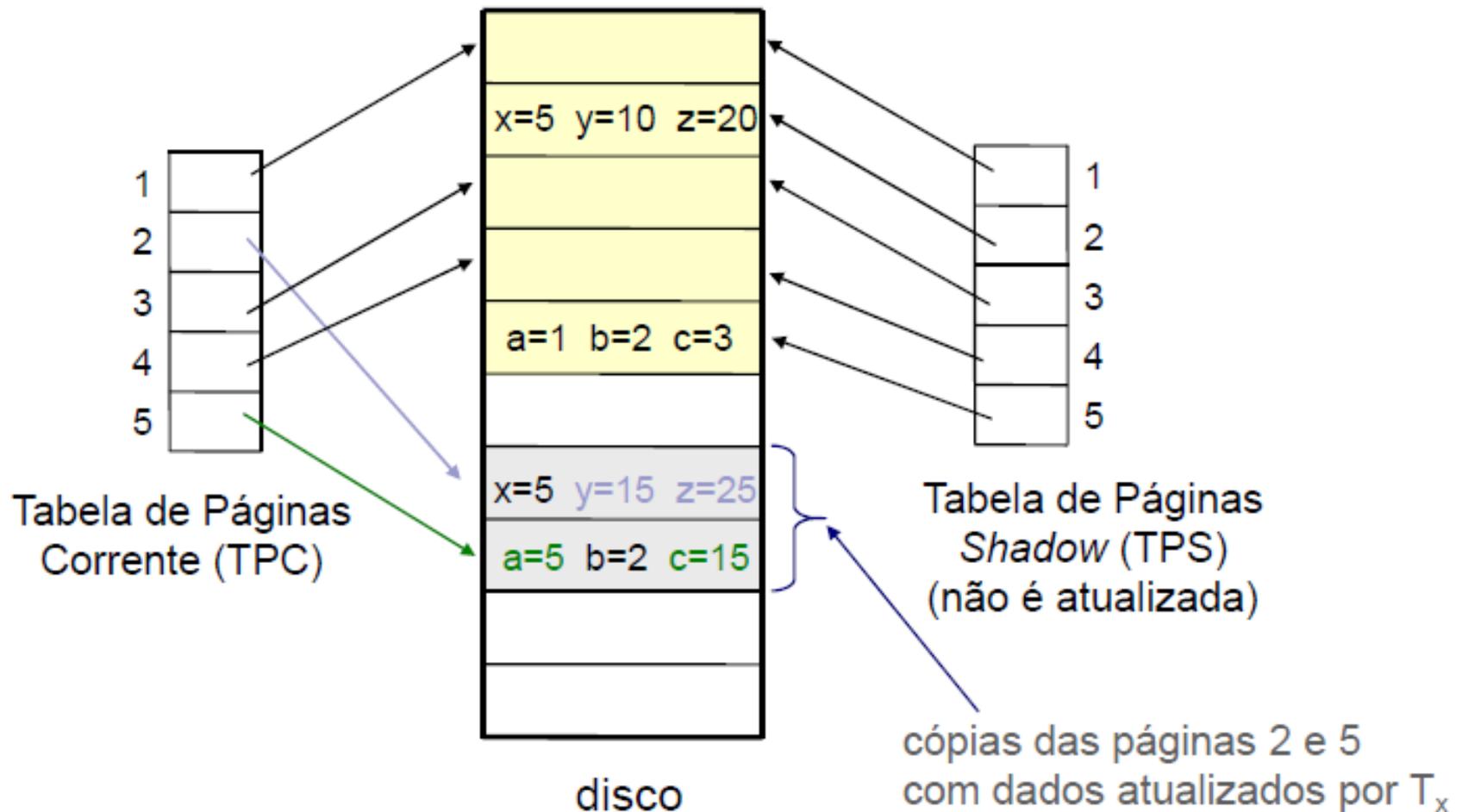
- ▶ Supõe a existência de uma tabela de blocos (páginas) de disco que mantém dados do BD (TPC)'
- ▶ A TPC é copiada para uma TPS a cada nova transação  $T_i$ 
  - Páginas atualizadas por  $T_i$  são copiadas para novas páginas de disco e TPC é atualizada;
  - TPS não é atualizada enquanto  $T_i$  está ativa.
- ▶ Em caso de falha de  $T_i$ , TPC é descartada e TPS torna-se a TPC
  - Não é preciso acessar o BD para realizar restaurações;
- ▶ Técnica NO-UNDO/NO-REDO

# Exemplo de Paginação Shadow



- ▶ Tabelas de páginas Shadow e corrente depois da escrita da página 4.

# Exemplo de Paginação Shadow



A técnica de recuperação de falhas com base em paginação sombra

- a) constrói um catálogo com um número  $n$  de entradas, onde a  $i$ -ésima entrada aponta para a  $i$ -ésima página do banco de dados em disco.
- b) mantém uma tabela shadow, que aponta para as páginas alteradas pela transação, enquanto a tabela original aponta para as páginas originais do banco de dados.
- c) considera que o banco de dados é composto por um número de páginas de tamanho fixo ou variável (quando as relações possuem campos BLOB ou CLOB).
- d) possui um catálogo que sempre será mantido em memória principal durante seu uso, independente do tamanho da base de dados.
- e) cria uma estrutura de log em disco, que armazena apenas as modificações nas páginas realizadas pelas transações nos elementos do banco de dados.

A técnica de recuperação de falhas com base em paginação sombra

- 
- a) constrói um catálogo com um número  $n$  de entradas, onde a  $i$ -ésima entrada aponta para a  $i$ -ésima página do banco de dados em disco.
  - b) mantém uma tabela shadow, que aponta para as páginas alteradas pela transação, enquanto a tabela original aponta para as páginas originais do banco de dados.
  - c) considera que o banco de dados é composto por um número de páginas de tamanho fixo ou variável (quando as relações possuem campos BLOB ou CLOB).
  - d) possui um catálogo que sempre será mantido em memória principal durante seu uso, independente do tamanho da base de dados.
  - e) cria uma estrutura de log em disco, que armazena apenas as modificações nas páginas realizadas pelas transações nos elementos do banco de dados.

A paginação shadow (sombra) é um esquema de recuperação utilizado em banco de dados que possibilita manter as informações originais alteradas por uma transação que foi cancelada. Esse esquema de recuperação é caracterizado por

- a) considerar que o banco de dados é composto de páginas ou blocos de disco de tamanho variável.
- b) necessitar do log (histórico) do SGBD se utilizado em ambiente monousuário.
- c) copiar o catálogo corrente, cujas entradas apontam para as páginas em disco mais recentes, para o catálogo shadow no início da transação.
- d) modificar o catálogo shadow, durante a execução da transação, para apontar para os blocos de disco contendo os dados alterados e incluídos.
- e) executar operações do tipo UNDO no cancelamento da transação.

A paginação shadow (sombra) é um esquema de recuperação utilizado em banco de dados que possibilita manter as informações originais alteradas por uma transação que foi cancelada. Esse esquema de recuperação é caracterizado por

- a) considerar que o banco de dados é composto de páginas ou blocos de disco de tamanho variável.
- b) necessitar do log (histórico) do SGBD se utilizado em ambiente monousuário.
-  c) copiar o catálogo corrente, cujas entradas apontam para as páginas em disco mais recentes, para o catálogo shadow no início da transação.
- d) modificar o catálogo shadow, durante a execução da transação, para apontar para os blocos de disco contendo os dados alterados e incluídos.
- e) executar operações do tipo UNDO no cancelamento da transação.

# Q26 – CESPE – MPE PI – 2012

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.

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.

# Q26 – CESPE – MPE PI – 2012

A respeito de administração de banco de dados relacionais;

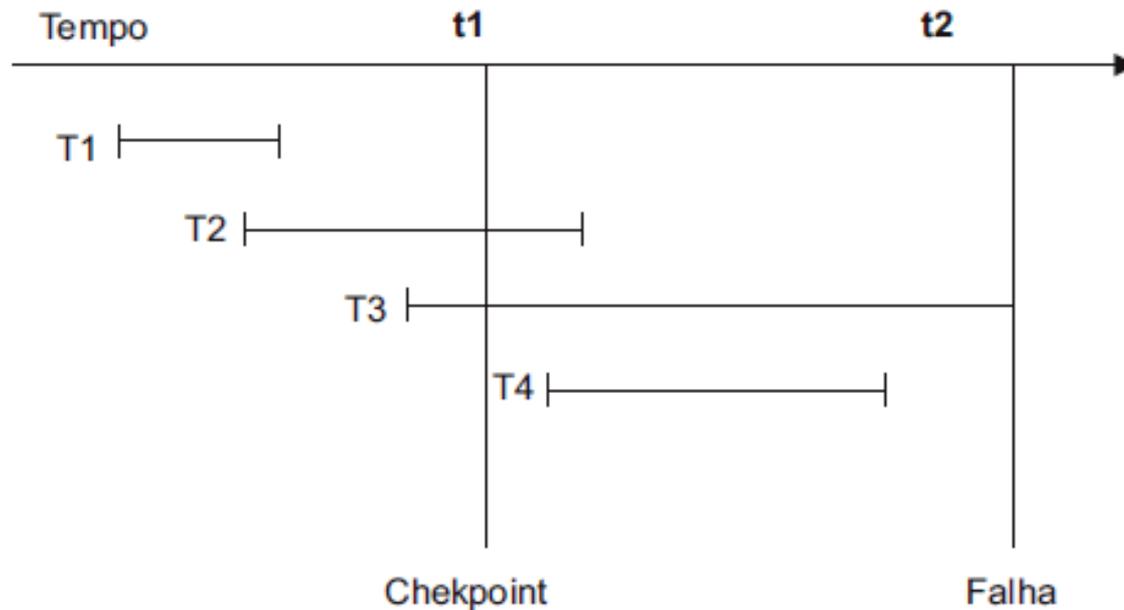
administração de usuários e perfis de acesso; controle de proteção, integridade e concorrência; restauração de dados; tolerância a falhas e continuidade de operação e otimização de desempenho, julgue os itens de 85 a 87.

**E**

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.

## Q27 – CESGRANRIO – BR Distribuidora – INFRA – 2011

A figura abaixo mostra quatro categorias de transações. A linha superior refere-se ao tempo que flui da esquerda para a direita. O tempo  $t_1$  marca um ponto de verificação (checkpoint), e no tempo  $t_2$  ocorre uma falha no sistema.

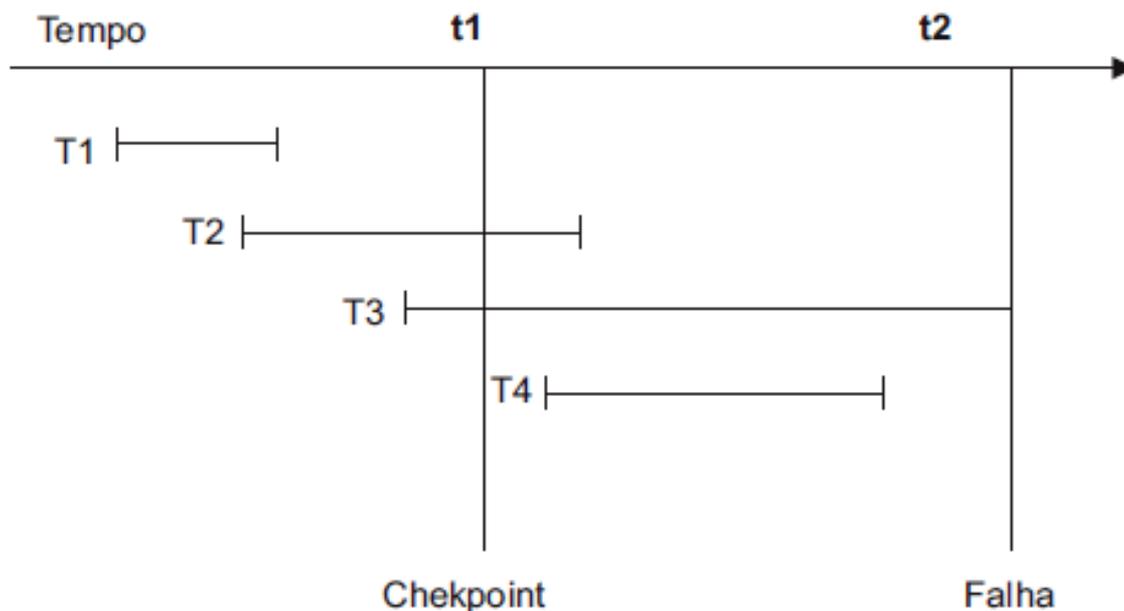


Quando o sistema for reinicializado, as transações do(s) tipo(s)

- (A) T3 terão que ser refeitas.
- (B) T2 e T3 terão que ser desfeitas.
- (C) T1 e T2 terão que ser refeitas.
- (D) T3 e T4 terão que ser desfeitas.
- (E) T2 e T4 terão que ser refeitas.

## Q27 – CESGRANRIO – BR Distribuidora – INFRA – 2011

A figura abaixo mostra quatro categorias de transações. A linha superior refere-se ao tempo que flui da esquerda para a direita. O tempo  $t_1$  marca um ponto de verificação (checkpoint), e no tempo  $t_2$  ocorre uma falha no sistema.

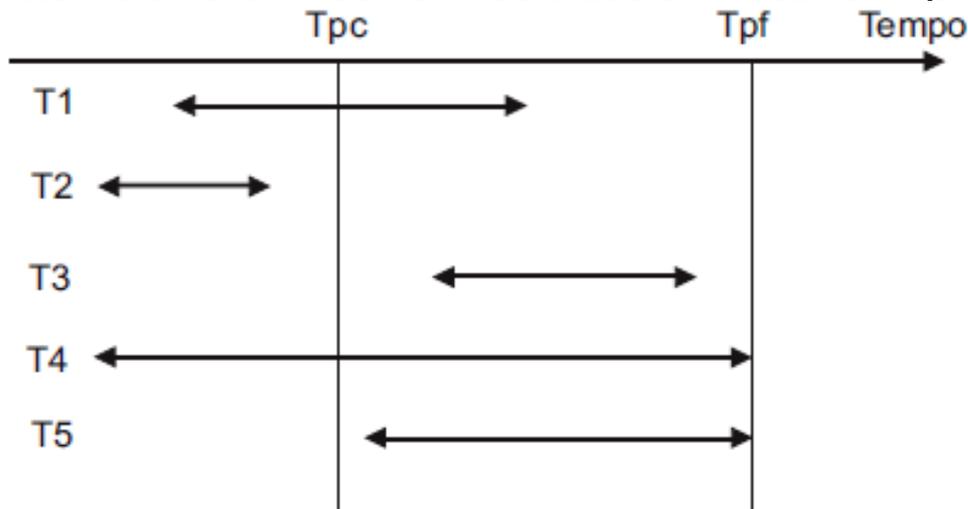


Quando o sistema for reinicializado, as transações do(s) tipo(s)

- (A) T3 terão que ser refeitas.
- (B) T2 e T3 terão que ser desfeitas.
- (C) T1 e T2 terão que ser refeitas.
- (D) T3 e T4 terão que ser desfeitas.
- (E) T2 e T4 terão que ser refeitas.

## Q28 – CESGRANRIO – Petrobras – INFRA – 2010

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  $T_{pc}$  é marcado um checkpoint, em  $T_{pf}$  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  $T_{pf}$ .

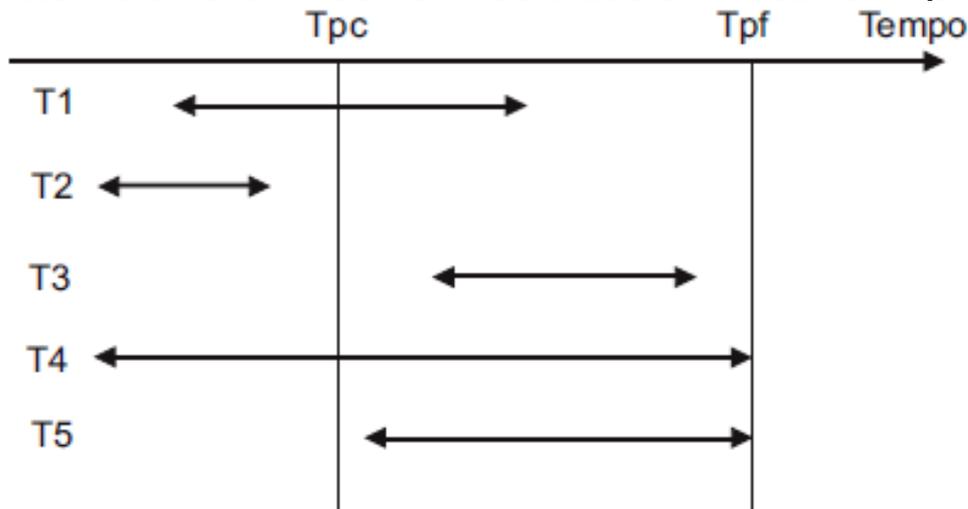


Após a falha ocorrer, o sistema é reinicializado e passa por um processo de recuperação até atingir um estado correto. No contexto apresentado, somente devem ser desfeitas as transações

- (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  $T_{pf}$ .

## Q28 – CESGRANRIO – Petrobras – INFRA – 2010

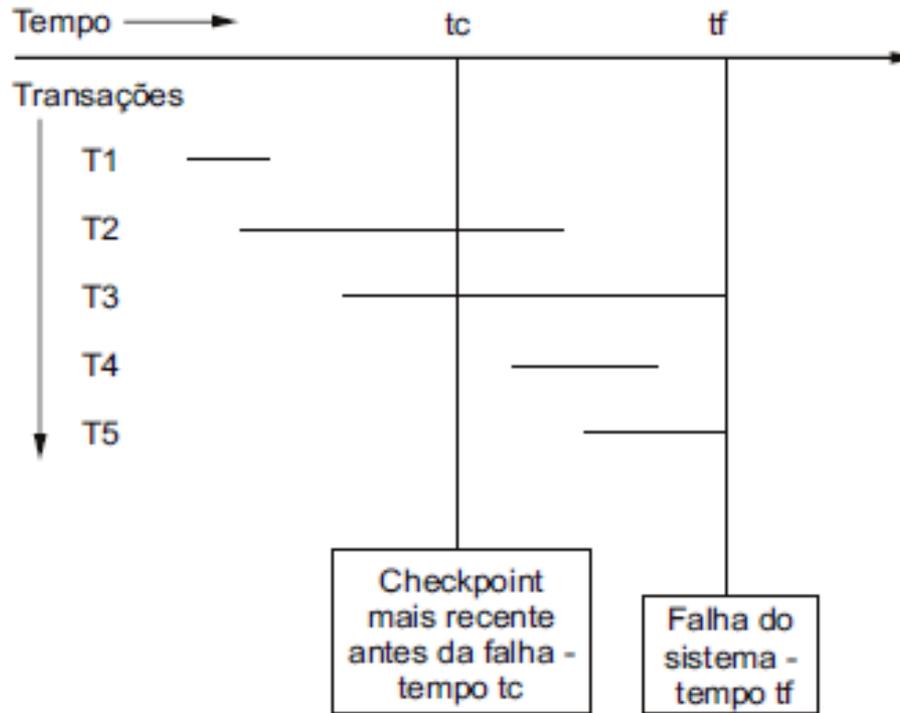
A figura abaixo apresenta a execução de cinco tipos de transações ao longo do tempo em um sistema gerenciador de bancos de dados cujas transações apresentam as propriedades ACID. No tempo  $T_{pc}$  é marcado um checkpoint, em  $T_{pf}$  há uma falha de sistema, as transações do tipo T1, T2 e T3 são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



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  $T_{pf}$ .

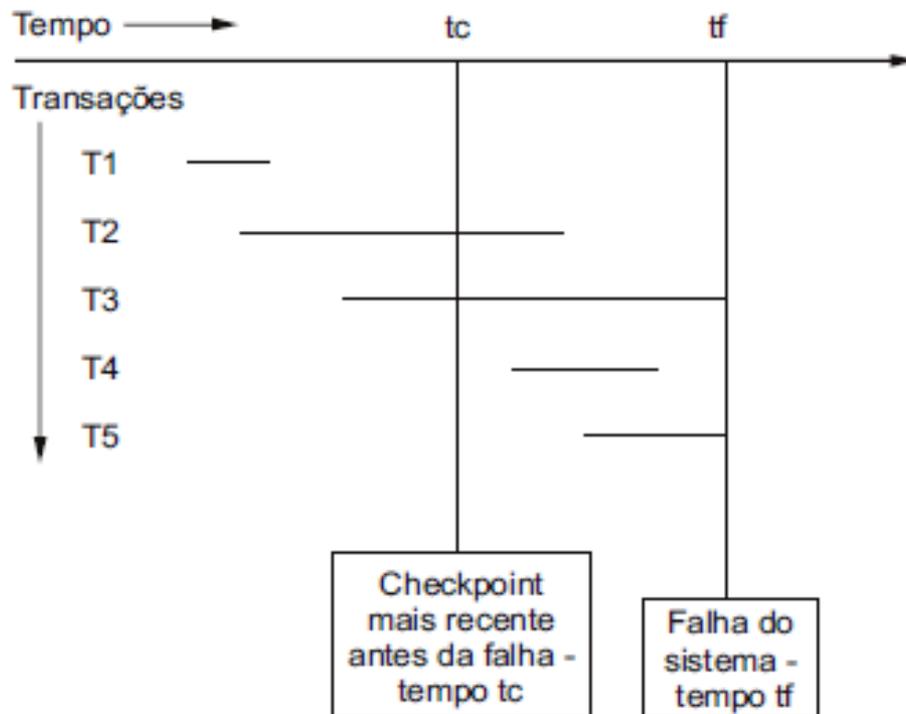
## Q29 – FCC – TCESP – 2009



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).

## Q29 - FCC - TCESP - 2009



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).

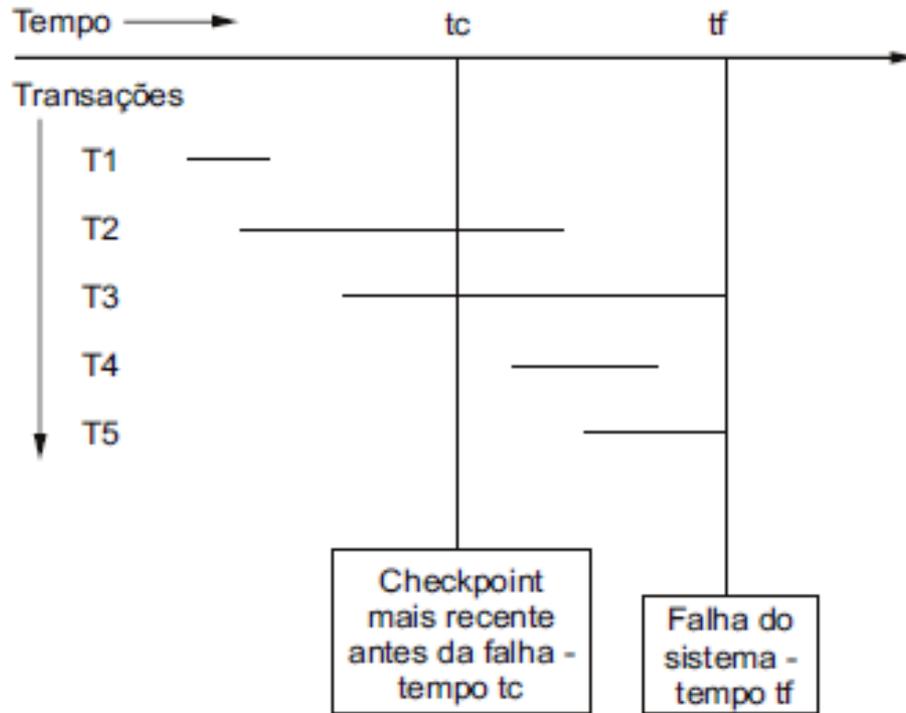
(B) T3 (D), T4 (D), T1 (R) e T2 (R).

(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).

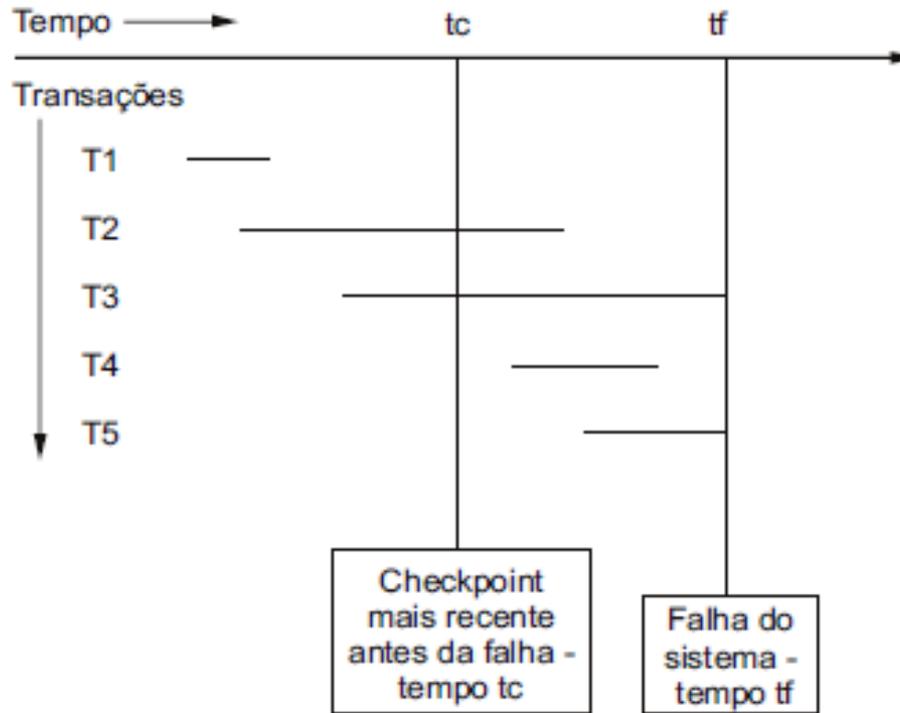
# Q30 - FCC - TCESP - 2009



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.

# Q30 - FCC - TCESP - 2009



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.

## Q31 – COVEST–COPSET – UFPE – 2010

A estrutura que mantém a informação sobre as alterações que foram aplicadas aos itens de dados a partir de transações e que é utilizada para restaurar um banco de dados para o estado de consistência mais recente é denominada de:

- a) catálogo.
-  b) Log.
- c) Cache.
- d) Checkpoint.
- e) LRU.

## Q32 – MPE–RS – 2012

A recuperação de transações que falharam indica que o banco de dados será restaurado para o estado de consistência mais recente, exatamente como antes do momento da ocorrência da falha. Para isso, o sistema deve manter informações sobre as alterações que foram aplicadas aos itens de dados pelas várias transações. Essas informações são armazenadas

- a) no diagrama entidade–relacionamento.
- b) no código armazenado.
- c) na consulta SQL.
-  d) no log (histórico) do sistema
- e) na tabela hashing.

## Q33 – TRT-10 – CESPE – 2012

- E** 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 roll back.

## Q34 – BASA BD – CESPE – 2010

- C** Uma transação do tipo dirty read lê todos os registros, não importando se estão sendo modificados ou se ainda não houve um commit.

# Q35 – FCC – TCESP – 2010

De acordo com C. J. Date, o critério de correção geralmente aceito para a execução intercalada de determinado conjunto de transações é

- a) a seriabilidade.
- b) a recuperação.
- c) o impasse.
- d) o escalonamento.
- e) o isolamento.

## Q36 – CESGRANRIO – PETROBRAS – PROCESSOS NEGÓCIOS – 2008

A execução de um conjunto de transações em um banco de dados é chamado de escalonamento. O critério para um escalonamento ser considerado correto é a(o)

- a) serializabilidade.
- b) durabilidade.
- c) leitura limpa (clean read).
- d) deadlock ausente (missing deadlock).
- e) commit alcançado (reached commit).

Um banco de dados possui dois objetos, O1 e O2. A transação T1 lê os objetos O1 e O2 e, então, grava os objetos O1 e O2. A transação T2 lê os objetos O1 e O2 e, então, grava o objeto O1.

O plano de execução que se inicia por:

T1:R(O1), T1R(O2), T1:W(O1), T2:R(O1) ...

Apresenta

- a) ausência de conflitos
-  b) um conflito gravação-leitura
- c) um conflito gravação-gravação
- d) um conflito leitura-gravação
- e) um conflito leitura-leitura

**Q38** – Quais as propriedades ACID das transações que um SGDB relacional multiusuário deve garantir?

- (A) Armazenamento, Consistência, Independência e Durabilidade
- (B) Armazenamento, Consistência, Isolamento e Determinação
-  (C) Atomicidade, Consistência, Isolamento e Durabilidade
- (D) Atomicidade, Confiabilidade, Isolamento e Durabilidade
- (E) Atomicidade, Confiabilidade, Independência e Determinação

**Q39** – As transações em um banco de dados devem possuir algumas propriedades, conhecidas como propriedades ACID.

Nesse contexto, a propriedade relacionada à letra I é a(o)

- (A) independência de dados
- (B) integridade referencial
- (C) inconsistência
- (D) identificação
-  (E) isolamento

## Q40 – FCC – MPERNN – 2012

Uma transação é uma sequência de operações executadas como uma única unidade lógica de trabalho. Uma unidade lógica de trabalho deve mostrar quatro propriedades para que seja qualificada como uma transação, sendo elas atomicidade, consistência, isolamento e

- (A) durabilidade.
- (B) rastreabilidade.
- (C) segurança.
- (D) visualização.
- (E) modularidade.

# Q41 – FCC – MPERNN – 2012

31. Em relação à execução de uma transação em um banco de dados, considere:

Para que uma transação seja efetivada, todas as ações que compõem a respectiva unidade de trabalho devem ser concluídas com sucesso. Caso contrário, a ação que constituiu falha e a transação devem ser desfeitas.

A afirmação refere-se a uma das quatro propriedades da integridade de uma transação, denominada:

-  (A) Atomicidade.
- (B) Isolamento.
- (C) Durabilidade.
- (D) Consistência.
- (E) Efetividade.

# Gabarito

Q16 – B

Q17 – C

Q18 – B

Q19 – D

Q20 – D

Q21 – E

Q22 – D

Q23 – D

Q24 – A

Q25 – C

Q26 – E

Q27 – E

Q28 – B

Q29 – C

Q30 – D

Q31 – B

Q32 – D

Q33 – E

Q34 – C

Q35 – A

Q36 – A

Q37 – B

Q38 – C

Q39 – E

Q40 – A

Q41 – A