



UNIVERSIDADE FEDERAL DE ITAJUBÁ

Banco de Dados II

COM 231

Controle de Concorrência

Vanessa Cristina Oliveira de Souza



UNIVERSIDADE FEDERAL DE ITAJUBÁ

TRATAMENTO DE DEADLOCK



Deadlock

- Um sistema está em um estado de impasse se houver um conjunto de transações tal que cada transação no conjunto está esperando por outra transação no conjunto.



Condições para haver Deadlock

- Exclusão mútua de recursos
 - Ou o recurso está sendo utilizado por algum processo, ou está disponível
- Uso e espera (*hold and wait*)
 - Processos que já possuem algum recurso podem requerer outros para finalizar a tarefa
- Espera Circular
 - Encadeamento circular de dois ou mais processos
- Não-preempção
 - Recursos já alocados não podem ser retirados do processo que os alocou; somente o processo que alocou o recurso pode liberá-lo.



Granularidade de Bloqueio

- A granularidade de bloqueio indica o nível de utilização do bloqueio
 - ☐ Nível de banco de dados
 - ☐ Nível de tabela
 - ☐ Nível de linha
 - ☐ Nível de atributo



Deadlock

- O tratamento de DeadLock pode ser feito de duas maneiras:
 - ☐ Prevenção de Impasse
 - ☐ Detecção e Recuperação de Impasse



Deadlock

■ Prevenção de Impasse

- ☐ Evita os deadlocks antes que eles ocorram
- ☐ Preferível se a probabilidade de ocorrerem deadlocks for muito alta

■ Detecção e Recuperação de deadlock

- ☐ Não evita os deadlocks, mas os detecta e impede o bloqueio indefinido das transações envolvidas
- ☐ Mais eficiente se ocorrerem poucos deadlocks

- O Rollback pode ser necessário independentemente da técnica utilizada



Prevenção de Deadlock

- Existem duas técnicas:
 1. **Assegura que nenhuma espera cíclica poderá ocorrer**
 - Exige que a transação bloqueie todos os itens antes de iniciar
 - Ordena os itens de dados e exige que uma transação bloqueie itens somente em uma sequência consistente com a ordenação.



Prevenção de Deadlock

- Existem duas técnicas:

- 2. Usar preempção e *rollbacks* de transação

- Esquema esperar-morrer (*wait-die*)

- Apenas as transações mais velhas (timestamp) têm permissão para esperar outra transação desbloquear um item de dado
 - As demais são abortadas (*rollback*)

- Esquema ferir-esperar (*wound-wait*)

- Apenas as transações mais novas(timestamp) têm permissão para esperar outra transação desbloquear um item de dado
 - As demais são abortadas (*rollback*)



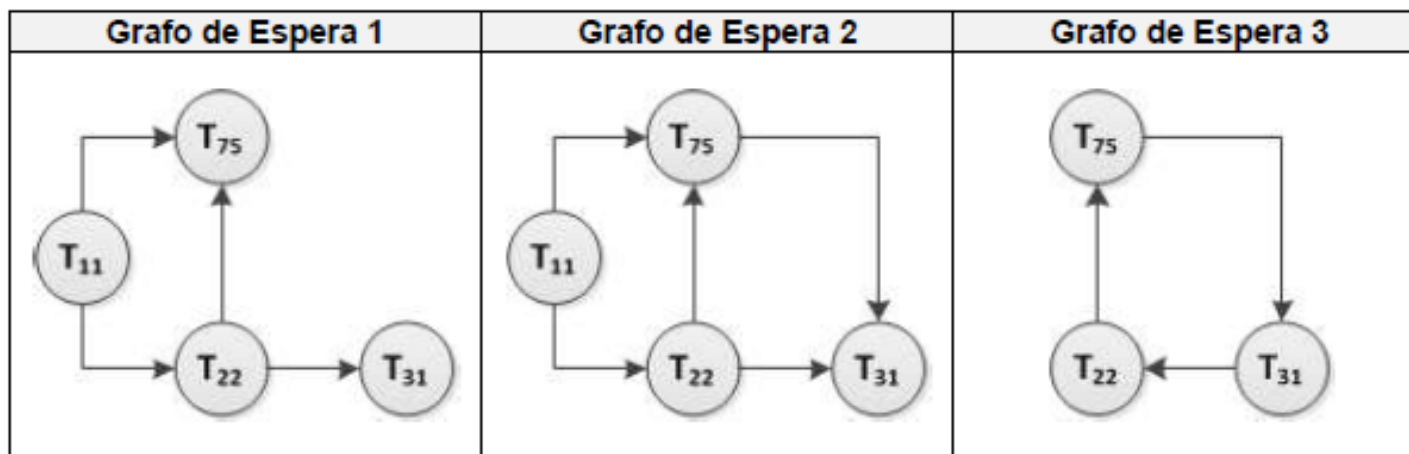
Detecção e Recuperação de Deadlock

- O sistema precisa processar as duas etapas:

1. **Detectar se o sistema está em *deadlock***

- Grafo de Espera

- ☐ Se houver ciclos no grafo, o sistema está em deadlock





Detecção e Recuperação de Deadlock

- O sistema precisa processar as duas etapas:

2. Recuperar o Sistema

- Selecionar a vítima
 - Livelock
 - O sistema tem dificuldades para selecionar a vítima
- Fazer o *Rollback*
 - É preciso reverter aquelas transações que incorrerão em custo mínimo.
 - Rollback parcial



Detecção e Recuperação de Deadlock

- Esse sistema pode levar uma transação a um estado chamado Estagnação (*starvation*)
 - Uma mesma transação sempre é escolhida como vítima, sendo sempre adiada, de forma que nunca é executada.



Testando o Deadlock

TRANSAÇÃO 1		TRANSAÇÃO 2
1	START TRANSACTION	
2		START TRANSACTION
3	UPDATE northwind.categories SET categoryid=15 WHERE categoryid=2;	
4		UPDATE northwind.products SET categoryid=15 WHERE categoryid=2;
5	UPDATE northwind.products SET categoryid=15 WHERE categoryid=2;	
6		SELECT DISTINCT categoryid FROM northwind.products;
7		UPDATE northwind.categories SET description='concorrência ativa' WHERE categoryid = 4;
8		COMMIT
9	COMMIT	



Testando o Deadlock

TRANSAÇÃO 1		TRANSAÇÃO 2
1		START TRANSACTION
2	START TRANSACTION	
3		UPDATE northwind.products SET categoryid=15 WHERE categoryid=2;
4	UPDATE northwind.categories SET categoryid=15 WHERE categoryid=2;	
5		SELECT DISTINCT categoryid FROM northwind.products;
6	UPDATE northwind.products SET categoryid=15 WHERE categoryid=2;	
7		UPDATE northwind.categories SET description='concorrência ativa' WHERE categoryid = 4;
8		COMMIT
9	COMMIT	



UNIVERSIDADE FEDERAL DE ITAJUBÁ

OUTRAS TÉCNICAS DE CONTROLE DE CONCORRÊNCIA



Controle de Concorrência

■ Multiversão

- *Multiversion Concurrency Control, MVCC*
- Cópias do mesmo **registro** para um conjunto de transações
- Transações divididas entre transações de leitura e de escrita
 - Transações de escrita criam cópias do dado
 - Transações de leitura só leem dados comitados
 - Operações de leitura **NÃO** bloqueiam operações de escrita e vice-versa.
 - Operações que alteram a estrutura da tabela precisarão bloquear qualquer operação de leitura/escrita concomitante.



Controle de Concorrência

■ Multiversão

- ☐ *Multiversion Concurrency Control, MVCC*
- ☐ Cópias do mesmo **registro** para um conjunto de transações
- ☐ Para garantir serializabilidade
 - Ordenação por *timestamp* em múltipla versão
 - Bloqueio de duas fases em múltipla versão



Controle de Concorrência

■ Isolamento de Instantâneo (*Snapshot*)

- Dá a uma transação uma “fotografia” do banco de dados no momento em que começa sua execução.
- Em seguida, ele opera sobre esse instantâneo em completo isolamento de transações concorrentes.
- As atualizações são feitas em um ‘espaço privado’
- As transações precisam, posteriormente, validar as atualização para não acontecer atualização perdida.
 - Primeiro confirmador vence
 - Rollback
 - Primeiro atualizador vence
 - Bloqueio



Controle de Concorrência

- *MVCC is the method PostgreSQL uses to handle data consistency when multiple processes are accessing the same table.*
- Para garantir a integridade dos dados durante a alteração da estrutura da tabela, o PostgreSQL aplica bloqueios no nível de tabela de forma automática.



Controle de Concorrência

Transaction Status

pg_clog

XID Status flags

028	0	0	0	1	0	0	1	0
024	1	0	1	0	0	0	0	0
020	1	0	1	0	0	1	0	0
016	0	0	0	0	0	0	1	0
012	0	0	0	1	0	1	1	0
008	1	0	1	0	0	0	1	0
004	1	0	1	0	0	0	0	0
000	1	0	0	1	0	0	1	0

Transaction Id (XID)

00 In Progress
01 Aborted
10 Committed

Ordenação por timestamp
em múltipla versão

Tuple	Creation XID: 15	Expiration XID: 27
-------	------------------	--------------------

xmin

xmax



UNIVERSIDADE FEDERAL DE ITAJUBÁ

NÍVEIS DE ISOLAMENTO EM SQL



Níveis de Isolamento

- Os níveis de isolamento de transação define quais efeitos colaterais de simultaneidade são permitidos, como leituras sujas ou leituras fantasma.
 - Nível de isolamento mais relaxado melhora performance, mas não garante consistência. E vice-versa.
- Comando pertence à SQL Padrão, mas cada SGBD apresenta uma sintaxe diferente. A mais comum é :
 - SET TRANSACTION ISOLATION LEVEL



Níveis de Isolamento

- As transações podem ser definidas segundo os seguintes níveis de isolamento:
 - READ UNCOMMITTED - Leitura Não Confirmada
 - Permite a leitura de dados não comitados
 - leitura suja
 - Na prática, não emite nenhum tipo de bloqueio



Níveis de Isolamento

- As transações podem ser definidas segundo os seguintes níveis de isolamento:
 - READ COMMITTED – Leitura Confirmada
 - Transações concorrentes não podem ler dados que foram atualizados, mas ainda não foram comitados



Níveis de Isolamento

- As transações podem ser definidas segundo os seguintes níveis de isolamento:
 - REPEATABLE READ – Leitura Repetível
 - Transações concorrentes não podem ler dados que foram atualizados, mas ainda não foram comitados.
 - Nenhuma outra transação pode modificar dados que foram lidos pela transação atual até que a transação atual seja concluída.



Níveis de Isolamento

- As transações podem ser definidas segundo os seguintes níveis de isolamento:
 - SERIALIZABLE – Serializável
 - Transações concorrentes não podem ler dados que foram atualizados, mas ainda não foram comitados.
 - Nenhuma outra transação pode modificar dados que foram lidos pela transação atual até que a transação atual seja concluída.
 - Outras transações não podem inserir linhas novas com valores chave que estejam no intervalo de chaves lido por qualquer instrução da transação atual até que esta seja concluída.



Para Casa



- Ler o **capítulo 18** do livro Sistemas de Banco de Dados – 4^a ed.
 - Elmasri e Navathe