UNIVERSIDADE ESTADUAL DE SANTA CRUZ (UESC)

Criada pela Lei 6.344, de 05.12.1991,
e reorganizada pela Lei 6.898, de 18.08.1995 e
pela Lei 7.176, de 10.09.1997

CET091 – Banco de Dados II

Prof. Dr. Marcelo Ossamu Honda

Departamento de Ciências Exatas e Tecnológicas (DCET)
mohonda(at)nbcgib(.)uesc(.)br

Controle de Concorrência

Controle de Concorrência

- Várias transações podem ser executadas simultaneamente no banco de dados;
  - Necessidade de métodos para manter a coerência dos dados;
- Necessário que o sistema controle a interação entre as transações concorrentes;
  - Assegurar a seriação;
  - Esquemas de controle de concorrência;
    - Adiam operação;
    - Abortam a transação que emitiu a operação;

Protocolos Baseados em Bloqueio

Protocolo Baseados em Bloqueio

- Conjunto de regras que que indicam quando uma transação pode bloquear ou desbloquear cada um dos itens de dados no banco de dados;
- Bloqueios;
  - Modos como um item de dados pode ser bloqueado;
    - Compartilhado;
      - Pode ler, mas não pode escrever sobre o item de dados;
    - Exclusivo;
      - Pode ler e escrever sobre o item de dados;

Protocolo Baseados em Bloqueio

- Funcionamento básico;
  - Transação solicita um bloqueio;
    - Modo apropriado sobre o item de dados;
  - O Gerenciador de Controle de Transação;
    - Concede o bloqueio a transação;
  - A transação pode prosseguir com a operação;
    - Nem sempre uma transação desbloqueie um item de dados imediatamente após seu acesso final desse item de dados;
      - Para assegurar a seriação;

Situação de Impasse

- Impasse (deadlock);
  - Quando nenhuma das transações pode prosseguir com a execução normal;
  - O sistema precisa reverter uma das transações;
    - Desbloqueando os dados da transação;
  - A outra transação, teoricamente, pode prosseguir;

T_3	T_4
lock-X(B)	
read(B)	
$B := B - 50$	
write(B)	
	lock-S(A)
	read(A)
	lock-S(B)
lock-X(A)	

Situação de Impasse

- Impasse (deadlock);
  - É considerado um mal necessário;
    - Para evitar estados inconsistentes;
      - Que podem gerar estados inconsistentes, não tratados pelo sistema de banco de dados;
    - Impasses podem ser revertidos por rollback de transações;

Protocolo de Bloqueio

- Conjunto de regras indicando quando uma transação pode bloquear e desbloquear cada um dos itens de dados;
  - Restringem o número de Schedules possíveis;
    - O conjunto de todos esses Schedules é um subconjunto apropriado de todos os Schedules serializáveis possíveis;
    - Um Schedule é legal sob determinado protocolo de bloqueio se for um Schedule possível para um conjunto de transações que seguem as regras do protocolo de bloqueio;
    - Um protocolo de bloqueio assegura a seriação de conflito se e somente se todos os Schedules legais forem passíveis de seriação de conflito;

Concessão de Bloqueio

- Bloqueio pode ser concedido;
  - Quando uma transação solicita um bloqueio sobre um item de dados em determinado momento;
  - Nenhuma outra transação possui um bloqueio sobre o mesmo item de dados em um modo em conflito;

Concessão de Bloqueio

- Transação estagnada (starved);
  - Quando uma transação não pode prosseguir;

T_1	T_2	T_3	Gerenciador de Controle De Transação
	Solicita Bloqueio Modo Compartilhado (Q)		
			Grant de Bloqueio Modo Compartilhado (Q) para T_1
Solicita Bloqueio Modo Exclusivo (Q)			
		Solicita Bloqueio Modo Compartilhado (Q)	
	Desbloqueio Modo Compartilhado (Q)		
			Grant de Bloqueio Modo Compartilhado (Q) para T_2

Q = item de dado

Concessão de Bloqueio

- Evitar a estagnação;
  - Não haja outra transação mantendo um bloqueio sobre um item de dados em um modo que entra em conflito com o modo solicitado;
  - Não existe outra transação que esteja esperando por um bloqueio sobre um item de dados e que fez sua solicitação de bloqueio antes da transação solicitante;
  - Assim uma solicitação de bloqueio nunca será bloqueada por uma solicitação de bloqueio que for feita mais tarde;

Protocolo de Bloqueio em Duas Fases

- Permite que uma transação bloqueie um novo item de dados somente se a transação ainda não tiver desbloqueado qualquer item de dados;
  - Ponto de bloqueio da transação;
    - Ponto no schedule em que a transação obteve seu bloqueio final;
      - Final da fase de crescimento;
- O protocolo assegura a seriação;
  - Mas não liberdade de impasse;
- Na ausência de informações referentes a maneira como os itens de dados são acessados;

Protocolo de Bloqueio em Duas Fases

- O protocolo de bloqueio estrito em duas fases permite a liberação de bloqueios exclusivos somente no final da transação;
  - Para assegurar a facilidade de recuperação e inexistência de cascata dos Schedules resultantes;
- Libera todos os bloqueios somente no final da transação;

Protocolo de Bloqueio em Duas Fases

- Protocolo de Bloqueio Estrito em Duas Fases;
  - Requer que o protocolo seja de duas fases;
  - Requer que todos os bloqueios no modo exclusivo realizado por uma transação seja mantido até que essa transação seja confirmada;
    - Evitando que qualquer outra transação leia os dados;
- Protocolo de Bloqueio Rigoroso em Duas Fases;
  - Requer que todos os bloqueios sejam mantidos até que a transação seja confirmada;

Protocolo de Bloqueio em Duas Fases

- Conversões de bloqueio;
  - Upgrade;
    - Conversão do modo compartilhado para o exclusivo;
    - Pode ocorrer apenas na fase de crescimento;
  - Downgrade;
    - Conversão do modo exclusivo para o compartilhado;
    - Pode ocorrer apenas na fase de encolhimento;
- Utilizados nos Sistemas de Banco de Dados comerciais;
  - Bloqueio Estrito em Duas Fases;
  - Bloqueio Rigoroso em Duas Fases (com conversões de bloqueio);

Protocolo de Bloqueio em Duas Fases

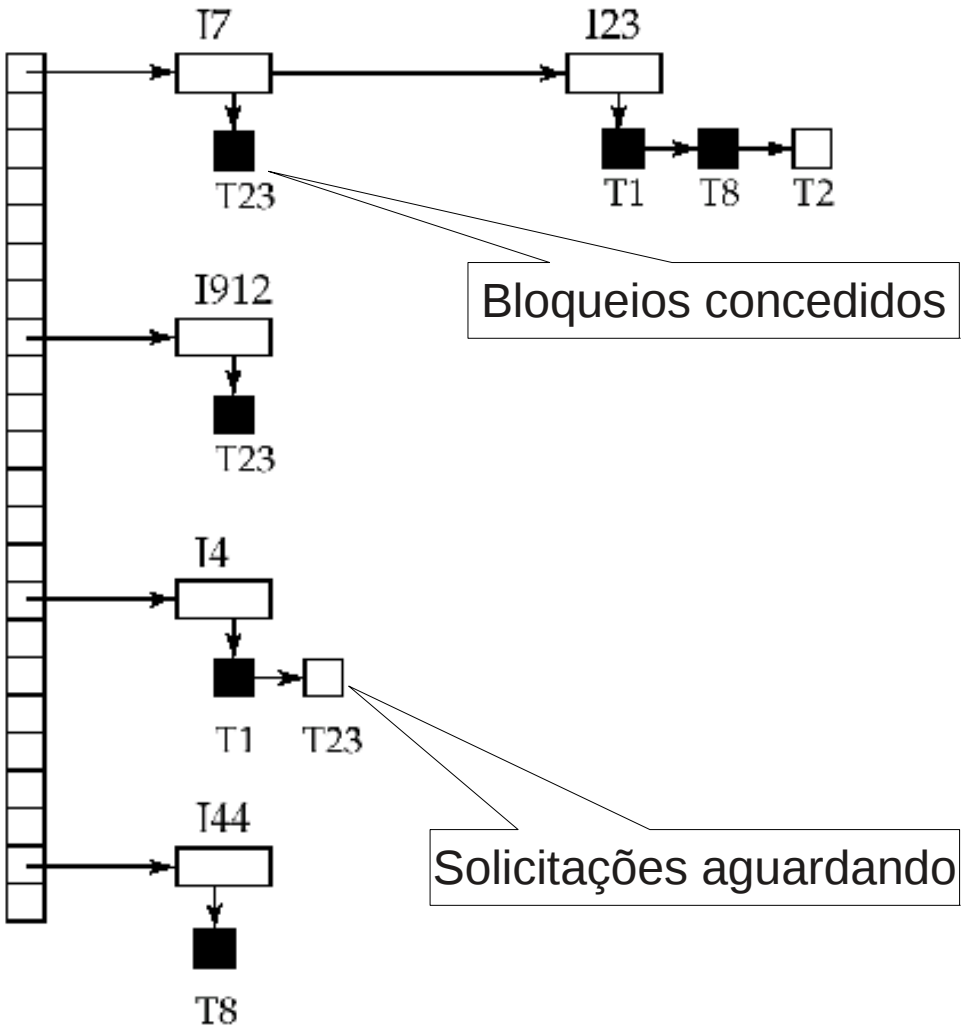
- Esquema simples;
  - Gera automaticamente as instruções apropriadas de bloqueio e desbloqueio para uma transação, com base nas solicitações de leitura e escrita da transação;
    - T_i emite uma operação read(Q);
      - O sistema emite uma instrução lock-s(Q), seguido pela instrução read(Q);
    - T_i emite uma operação write(Q);
      - O sistema verifica se T_i já mantem um bloqueio compartilhado;
        - O sistema emite uma instrução upgrade(Q), seguido pela instrução write(Q);
      - Caso contrário, o sistema emite uma instrução lock-x(Q), seguido pela instrução write(Q);
    - Todos os bloqueios obtidos por uma transação são desbloqueados depois que a transação é confirmada ou abortada;

Implementação do Bloqueio

- Gerenciador de Bloqueio;
  - Pode ser implementado com um processo que recebe mensagens das transações e envia mensagens de resposta;
    - Responde a mensagens de solicitação de bloqueio com mensagens de concessão de bloqueio ou com mensagens solicitando o rollback da transação;
    - As mensagens de desbloqueio exigem apenas uma confirmação em resposta;
      - Mas pode resultar em uma mensagem de concessão para outra transação aguardando;
    - O gerenciador de bloqueio utiliza uma estrutura de dados para manter o controle;

Implementação do Bloqueio

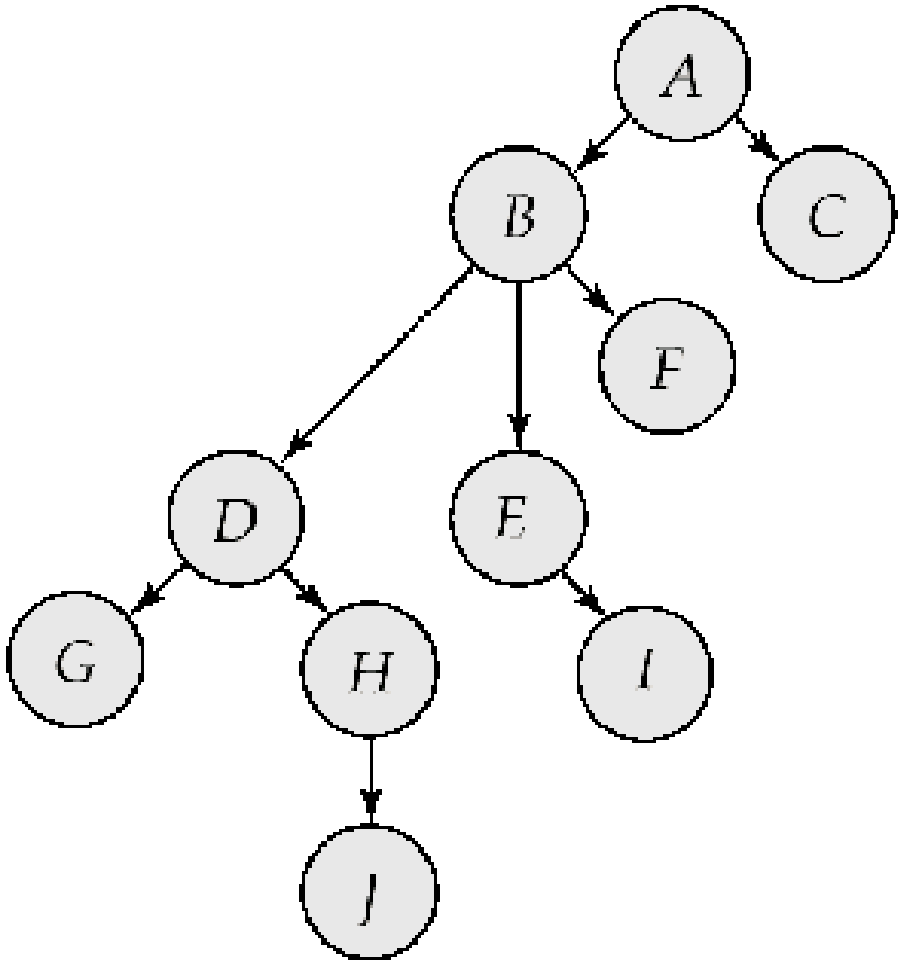
- Gerenciador de Bloqueio;
  - Lista interligada de registros;
    - Para cada item de dados que está atualmente bloqueado;
    - Uma para cada solicitação, na ordem de chegada;
  - Tabela hash;
    - Tabela de bloqueio;
    - Indexada sobre o nome de um item de dados, para encontrar a lista interligada para um item de dados;
    - Cada registro da lista interligada para um item de dados anota qual transação fez a solicitação e qual o modo de bloqueio solicitado e se a solicitação atualmente foi concedida;



Protocolos Baseados em Gráfico

- Protocolo em Duas Fases;
  - Necessário e suficiente para garantir a seriação na ausência de informações com relação a maneira em que os itens de dados são acessados;
- Protocolos não em Duas Fases;
  - Informações adicionais sobre como cada transação acessará o banco de dados;
  - Vários modelos, que diferem na quantidade de informações fornecidas;
    - O modelo mais simples, exige conhecimento sobre a ordem em que os itens do banco de dados serão acessados;

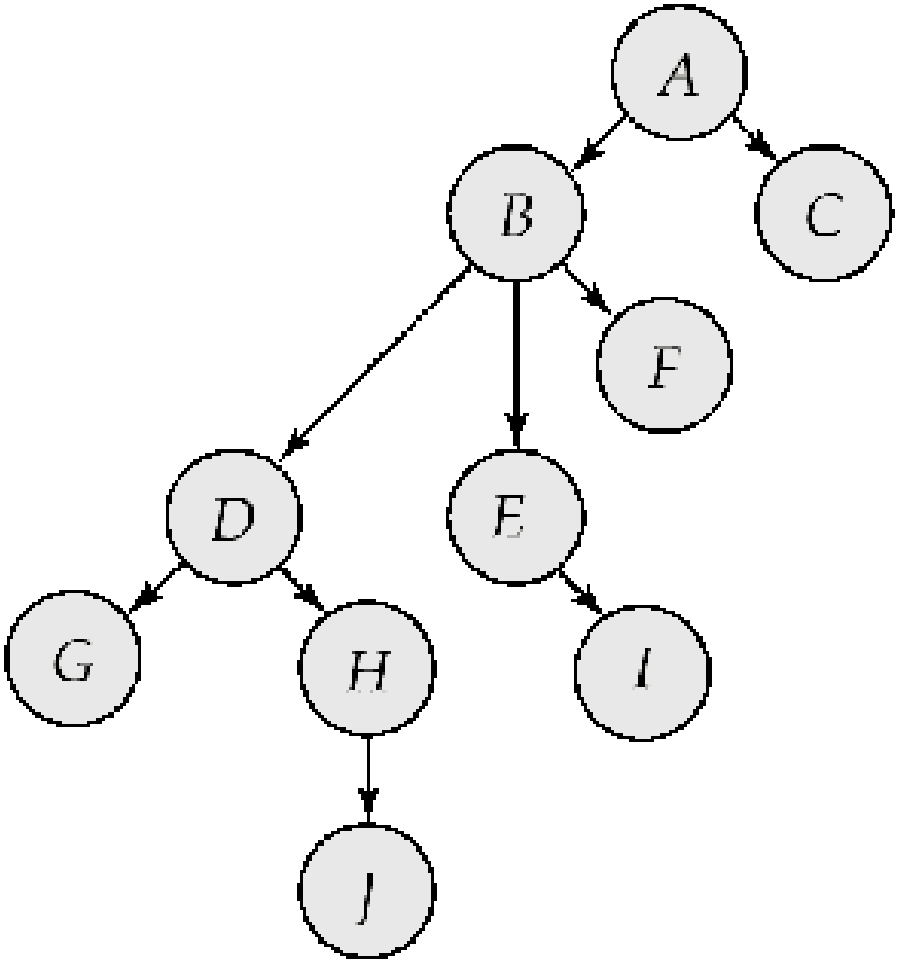
Protocolos Baseados em Gráfico



- Gráfico de Banco de Dados;
  - Ordenação parcial;
    - Gráfico acíclico direcionado;
    - Organização lógica ou física do banco de dados;
    - Pode ser imposta unicamente para fins de controle de concorrência;
- Protocolo simples;
  - Protocolo de Árvore;
    - Utiliza apenas bloqueios exclusivos;

Protocolos Baseados em Gráfico

- Exemplo:
  - Transação T13;
    - lock-x(D);
    - lock-x(H);
    - unlock-x(D);
    - unlock-x(H);
  - Resumo:
    - O primeiro bloqueio pode ser em qualquer item de dados;
    - O próximo item bloqueado, somente pode ocorrer se o item de dados pai estiver bloqueado, pela transação;
    - Os itens de dados podem ser desbloqueados a qualquer momento;
    - Uma transação não pode bloquear/desbloquear novamente um item de dados;



Protocolos Baseados em Gráfico

- O protocolo de árvore;
  - Garante o seriação de conflito;
  - Assegura a liberdade de impasse;
  - Não garante a facilidade de recuperação e inexistência de cascata;
    - Alternativas;
      - Não permitir a liberação de bloqueios exclusivos até o final da transação;
        - Reduz a concorrência;
      - Dependência de commit;
        - Melhora a concorrência e facilidade de recuperação;

Protocolos Baseados em Gráfico

- Em relação ao Protocolo em Duas Fases;
  - Vantagens;
    - Livre de impasse;
    - Desbloqueio mais cedo;
  - Desvantagens;
    - Bloquear itens de dados que não acessa;
      - Sobrecarga de bloqueio (tempo de espera);
      - Diminuição da concorrência;
    - Pode ser necessário bloquear a raiz da árvore;

Protocolos Baseados em Timestamp

Timestamp

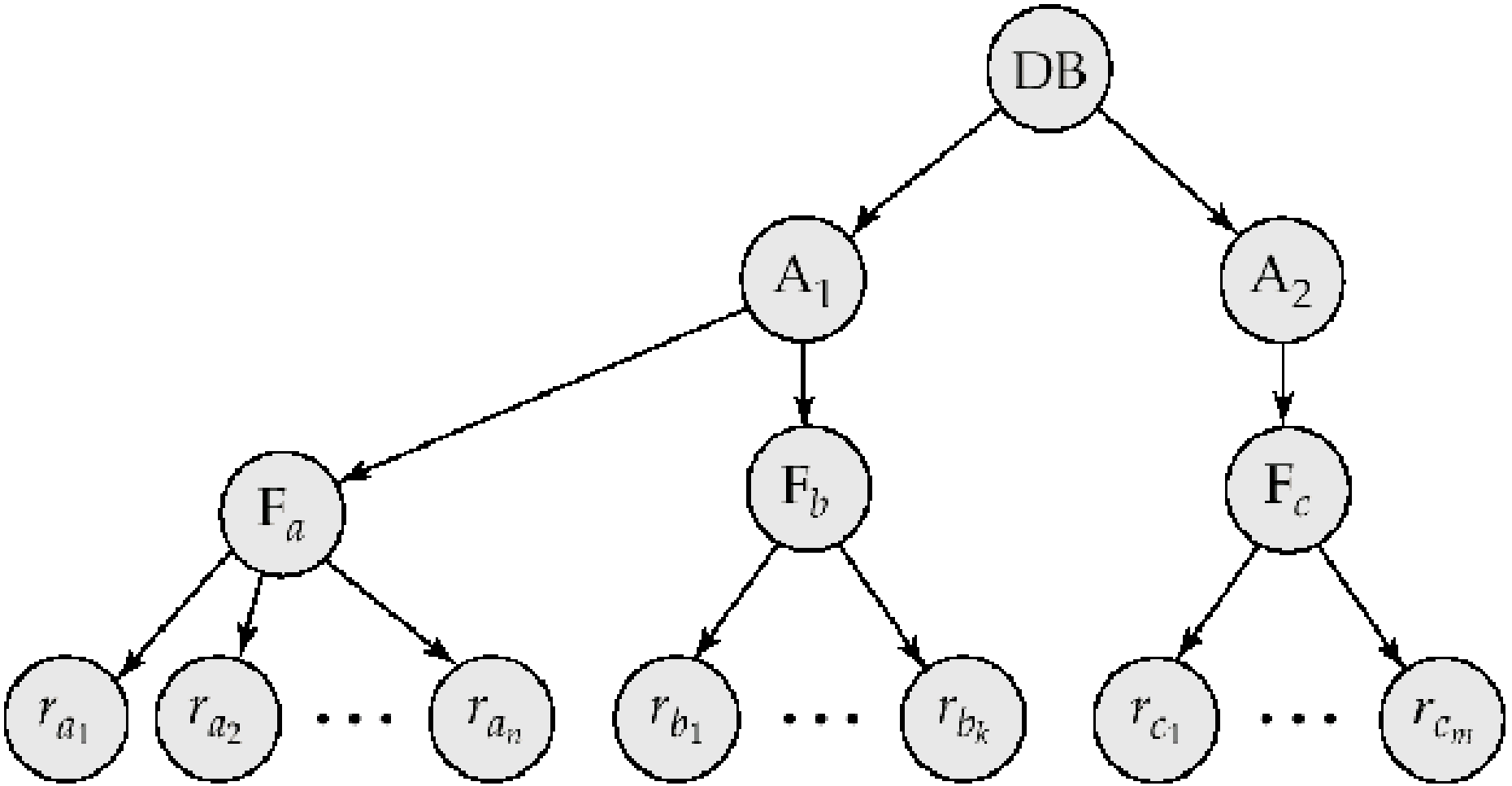
- Determinar a ordem de seriação através da ordenação entre as transações com antecedência;
  - A ordem é mantida realizado o rollback da transação que viola a ordem;

Granularidade Múltipla

Granularidade Múltipla

- Quando se pode agrupar diversos itens de dados e tratá-los como um item de dados agregado para fins de trabalho;
  - Resultando em vários níveis de granularidade;
  - Itens de dados menores são aninhados dentro dos maiores;
    - Hierarquia representada em uma árvore;
      - Bloqueios são adquiridos na ordem da raiz para a folha;
      - Liberados na ordem da folha para raiz;
  - Protocolo garante seriação;
    - Mas não está livre de impasses;

Granularidade Múltipla



Esquemas de Múltipla Versão

Esquemas de Múltipla Versão

- Uma nova versão de um item de dados é criado para cada transação que escreve esse item;
  - Na operação de leitura, o sistema deve selecionar uma das versões para ser lida;
    - Versão que garanta a seriação;
  - Timestamp:
    - Uma operação de escrita pode resultar em rollback da transação;
  - Bloqueio em duas fases;
    - Uma operação de escrita pode ter que esperar pelo bloqueio, ou gerar impasse;

Tratamento de Impasse

Tratamento de Impasse

- Um sistema está em estado de impasse se houver um conjunto de transações tal que cada transação no conjunto está esperando por outra transação no conjunto;
  - Conjunto de transações $\{T_0, T_1, ..., T_n\}$
    - T_0 aguarda por um item de dados que T_1 mantem;
    - T_1 aguarda por um item de dados que T_0 mantem;
    - Nenhuma das transações pode prosseguir em tal situação;
  - Solução, o sistema precisa reverter algumas das transações envolvidas no impasse;
    - Pode ser um rollback parcial de uma transação;
      - Revertido até o ponto que resolva o impasse;

Tratamento de Impasse

- Métodos para lidar com o impasse;
  - Prevenção de Impasse;
    - Método para garantir que o sistema nunca entre num estado de impasse;
    - Pode resultar em rollback;
    - Usando quando a probabilidade de um sistema entrar em um estado de impasse for alta;
  - Detecção e recuperação de impasse;
    - O sistema pode entrar num estado de impasse, mas aplique o método para recuperação;
    - Pode resultar em rollback;
    - Método mais eficiente;
    - Maior sobrecarga ao sistema;

Prevenção de Impasse

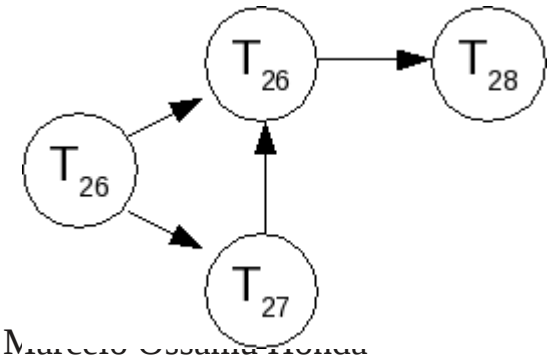
- Técnica simples;
  - Exige que cada transação bloqueie todos os seus itens de dados antes de iniciar a execução;
  - Desvantagens;
    - Difícil prever, antes que a transação inicie, que itens de dados precisam ser bloqueados;
    - Utilização baixa do itens de dados;

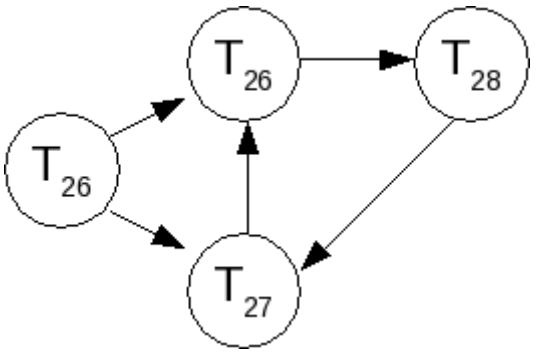
Prevenção de Impasse

- Técnica (preempção e rollback);
  - Uma transação pode apropriar dos dados utilizados por outra transação, forçando o rollback da segunda transação;
    - Timestamp exclusivo para cada transação;
      - O sistema usa esse timestamp somente para decidir se uma transação deverá esperar ou reverter;

Detecção de Impasse

- Realizado por um algoritmo que examina o estado do sistema periodicamente, para determinar se um impasse ocorreu;
- Gráfico de Espera (wait for);
  - Vértices e Arestas;
  - Impasse determinado por ciclo no gráfico;





Recuperação de Impasse

- Após a detecção de impasse é necessário que o sistema retorne a um estado consistente anterior;
  - Reverter uma ou mais transações;
  - Ações;
    - Seleção de uma vítima;
      - Determinar quais transações reverter (custo minimo);
    - Rollback;
      - Determinado a transação é preciso determinar até que ponto deve ser revertido;
    - Estagnação;
      - Processo de seleção determinar sempre a mesma vítima;

Operações de Inserção e Exclusão

Operações de Inserção e Exclusão

- Exclusão:
  - A transação excluindo a tupla deve ter um bloqueio exclusivo sobre a tupla a ser excluída;
- Inclusão:
  - Uma transação que insere uma nova tupla no banco de dados recebe o bloqueio exclusivo sobre a tupla;
  - Fenômeno fantasma:
    - Uma inserção entra em conflito lógico com uma consulta;
      - Mesmo que ambas não acessem uma tupla em comum;

Níveis de Consistência Fracos

Níveis de Consistência Fracos

- Pode ser utilizado quando a consistência dos resultados da consulta não é critica;
  - Os protocolos de seriação podem resultar em baixa concorrência para alguns tipos de aplicações;
  - Responsabilidade do desenvolvedor garantir a integridade do banco de dados;
- Consistência de grau dois;
  - Modos de bloqueio do protocolo duas fases;
    - Compartilhado e Exclusivo;
    - Bloqueios podem ser liberados/adquiridos um de cada vez;
    - Bloqueios exclusivos não podem ser liberados até que a transação confirme/aborte;
    - A seriação não é garantida;

Níveis de Consistência Fracos

- Estabilidade de cursor;
  - Forma de consistência de grau dois;
    - Para programas escritos nas linguagens host;
    - Percorrem as tuplas usando cursores;
  - Em vez de bloquear a relação inteira, a estabilidade do cursor garante:
    - A tupla processada atualmente pela iteração seja bloqueada no modo compartilhado;
    - Quaisquer tuplas modificadas sejam bloqueadas no modo exclusivo, até que a transação aborte/confirme;
    - Procura aumentar a concorrência e melhorar o desempenho do sistema;

Níveis de Consistência Fracos

- Níveis especificados pela SQL-92:
  - Passíveis de seriação;
  - Leitura repetitiva;
    - Somente registros confirmados podem ser lidos;
    - Entre duas leituras de um registro por uma mesma transação, nenhuma outra transação tenha permissão de atualizar o registro;
  - Leitura confirmada;
    - Somente registros confirmados podem ser lidos;
  - Leitura não confirmada;
    - Todos os registros podem ser lidos;

Concorrência em Estruturas de Índices

Concorrência em Estruturas de Índices

- Tratar o acesso a estrutura de índice como qualquer outra estrutura de banco de dados e aplicar as técnicas de controle de concorrência;
  - Baixo grau de concorrência;
    - Índices acessados com frequência, grande disputa de bloqueio;
  - Uma transação pode realizar uma pesquisa no índice duas vezes;
    - Determinar que a estrutura de índice mudou, nesse espaço de tempo;
    - Importante é que a consulta retorne o conjunto correto de tuplas;
  - Protocolo:
    - Caranguejo;
    - Bloqueio de árvore B-link;

PostgreSQL

Chapter 13. Concurrency Control

This chapter describes the behavior of the PostgreSQL database system when two or more sessions try to access the same data at the same time. The goals in that situation are to allow efficient access for all sessions while maintaining strict data integrity. Every developer of database applications should be familiar with the topics covered in this chapter.

13.1. Introduction

PostgreSQL provides a rich set of tools for developers to manage concurrent access to data. Internally, data consistency is maintained by using a multiversion model (Multiversion Concurrency Control, MVCC). This means that while querying a database each transaction sees a snapshot of data (a *database version*) as it was some time ago, regardless of the current state of the underlying data. This protects the transaction from viewing inconsistent data that could be caused by (other) concurrent transaction updates on the same data rows, providing *transaction isolation* for each database session. MVCC, by eschewing explicit locking methodologies of traditional database systems, minimizes lock contention in order to allow for reasonable performance in multiuser environments.

Referências

- Ramez Elmasri e Shamkant B, Navathe, Sistemas de Banco de Dados, Pearson Addison Wesley, 2005;
- Abraham Silverschatz, Henry F. Korth e S. Sudarshan, Sistema de Banco de Dados, Editora Campus, 2006;
- PostgreSQL 8.3.6 Documentation, by The PostgreSQL Global Development Group, Copyright © 1996-2008 The PostgreSQL Global Development Group;

Referências

- KORTH, H. F., SILBERSCHATZ, A. Sistema de Banco de Dados, Makron Books
- DATE, C. J. Introdução a Sistemas de Banco de Dados. Tradução da 7ª. Edição Americana. Editora Campus.
- DATE, C.J. Bancos de Dados, Tópicos Avançados, Editora Campus
- GARCIA-MOLINA, H. ULLMAN, J.D., WIDOM, J. Implementação de Sistemas de Bancos de Dados. Editora Campus
- Manuais Técnicos e Livros voltados para Bancos de Dados específicos.