

Transações

Bancos de Dados
Prof. Geraldo Corrêa

BD - Transações

- Banco de Dados não é um sistema único
- Pode ser acessado por várias pessoas de locais diferentes
- Exemplo:
 - Uma conta corrente conjunta pode ser acessada pelas 2 pessoas simultaneamente e de locais diferentes. Quem sacará primeiro?
- É para evitar problemas assim que existe a Transação

2

BD - Transações

- “Sistemas de processamento de transação são sistemas com grandes bancos de dados e centenas de usuários executando transações concorrentes no banco de dados” (ELMASRI e NAVATHE, 2005, p. 397)
- Uma transação é uma sequência de operações tratadas como um bloco único e indivisível. A execução parcial não é permitida.

3

BD - Transações

- O final da transação é indicado pelos comandos:
 - **Commit:** Quando a transação é encerrada com sucesso. Todas as modificações feitas durante a transação são efetuadas no banco de dados
 - **Rollback:** Quando a transação é encerrada com um erro. Todas as modificações feitas no banco de dados durante a transação são ignoradas, restaurando o banco ao seu estado anterior ao início da transação
- “Transação é uma unidade atômica de trabalho que estará completa ou não foi realizada” (ELMASRI e NAVATHE, 2005, p. 402)

4

BD - Transações

- A transação envolve as seguintes operações no banco de dados:
 - Inserção
 - Alteração
 - Exclusão
- Estas operações são executadas utilizando-se a linguagem SQL (Structure Query Language – Linguagem Estruturada de Consulta)
- Em linguagem SQL, as operações envolvidas na transação são, respectivamente: **Insert**, **Update** e **Delete**.

5

BD - Transações

- Os problemas mais comuns em transações são:
 - **Atualização perdida:** Ocorre quando duas transações são emitidas quase que simultaneamente. A segunda transação utiliza os dados antes que a primeira tenha efetivado as mudanças. Quando a segunda atualiza o Banco de Dados, as modificações da primeira transação são perdidas.
 - **Atualização Temporária:** Ocorre quando há um erro em uma transação e uma segunda acessa os dados antes que os valores originais sejam restaurados, usando assim dados incorretos.

6

BD - Transações

- Os problemas mais comuns em transações são:
 - **Sumário Incorreto:** Ocorre quando uma transação usa a função de sumarização enquanto uma outra ainda atualiza os dados
 - **Leitura sem repetição:** Ocorre quando uma transação deve ler um dado duas vezes e entre as leituras uma outra transação efetua uma modificação nele.

7

BD - Transações

- Algumas possibilidades para ocorrência de falhas durante uma transação:
 - Falha do computador (hardware, software ou rede)
 - Erro de transação ou sistema: Erros de lógica de programação, valores de parâmetros, etc
 - Erros locais ou condições de exceções detectadas pela transação
 - Falha de disco (de leitura ou gravação)
 - Problemas físicos e catástrofes: falta de energia, fogo, furto, etc

8

BD - Transações

- São possíveis estados de uma transação:
 - **begin_transaction:** Marca o início de uma transação, passando ao **estado ativo**
 - **read** ou **write:** São leituras ou gravações feitas enquanto a transação se encontra no estado ativo. Ao encerrar-se, passa ao **estado de efetivação parcial**
 - **commit_transaction:** Efetua no banco de dados as modificações feitas e não permite que sejam desfeitas, passando ao **estado de efetivação**
 - **rollback:** Operação contrária do COMMIT, retornando o banco de dados ao estado anterior ao início da transação, passando a transação ao **estado de falha**
 - **end_transaction:** Marca o fim da transação, passando ao **estado encerrado**

9

BD - Transações

- Propriedades das transações – ACID
 - **Atomicidade:** Uma transação deve ser executada por completo ou não ser executada. Quando há uma falha, as técnicas de recuperação devem desfazer totalmente as modificações feitas
 - **Consistência:** Um Banco de Dados é consistente quando satisfaz a restrições específicas, assim, deve continuar sendo consistente ao término de uma transação, cuja responsabilidade é do programador ou do SGBD
 - **Isolamento:** Um transação não pode sofrer interferência de transações concorrentes
 - **Durabilidade:** As atualizações feitas por uma transação devem persistir no Banco de Dados, independente da ocorrência de falhas.

10

BD - Transações

- Para controlar a ordem de execução das transações é criado um **plano de execução**, que determina a sequência das operações e das transações
- O plano de execução pode basear-se na **restaurabilidade** ou na **serialidade**
- A restaurabilidade garante que nenhuma transação T seja efetivada se existirem transações que ainda não efetivaram os dados lidos pela transação T

11

BD - Transações

- Na serialidade, as operações das transações são executadas consecutivamente, sem intercalações com operações de outras transações mas, desperdiça tempo de processamento de CPU
- Conflito
 - Duas operações de um plano estão em conflito se pertencem a transações diferentes, tentem acessar um mesmo item do banco de dados e, ao menos uma delas é um operação de gravação

12

BD - Transações

- Para controlar transações concorrentes utiliza-se técnicas de bloqueio
- O **bloqueio** (*lock*) indica o que pode ser aplicado em relação às operações que são executadas pelas transações. Deve haver um bloqueio para cada dado no banco de dados.
- Tipos de bloqueio
 - **Bloqueios compartilhados**
 - **Bloqueios exclusivos**

13

BD - Transações

- Operações de bloqueio
 - **Read_lock(X)** – “read_locked”
 - **Write_lock(X)** – “write_locked”
 - **Unlock(X)** – “unlocked”
- Elmasri e Navathe (2205) definem regras para utilização de bloqueio compartilhado/exclusivo:
 - Antes de ler um item deve garantir a operação *read_lock* ou *write_lock* dele
 - Antes de escrever em um item, deve garantir a operação *write_lock* dele
 - Após todas as operações de leitura e escrita deve garantir a operação de *unlock* dele

14

BD - Transações

- Elmasri e Navathe (2205) definem regras para utilização de bloqueio compartilhado/exclusivo:
 - Se uma operação já tem um bloqueio de leitura ou de escrita não fará a operação de *read_lock* dele
 - Se uma operação já tem um bloqueio de leitura ou de escrita não fará a operação de *write_lock* dele
 - Se uma operação NÃO tem um bloqueio de leitura ou de escrita não fará a operação de *unlock* dele

15

BD - Transações

- Deadlock
 - É a espera de uma transação por um item que está bloqueado por uma outra transação
 - Detecção do deadlock
 - Indicada para transações que tem poucos acessos aos mesmos itens simultaneamente
 - Quando o deadlock é detectado, algumas transações que o causam devem ser abortadas
 - O uso de *timeouts* pode ser utilizado para prevenir os *deadlocks*

16