

Universidade Presbiteriana Mackenzie



Banco de Dados – Aula 21

Transação

Profa. Elisângela Botelho Gracias

Faculdade de Computação e Informática

Transações

- Definição de transação
 - é uma unidade lógica de trabalho
 - ou seja, é uma sequência de várias operações sobre um banco de dados, cujo objetivo é transformar o banco de dados de um estado consistente para outro estado consistente, mesmo que nos passos intermediários o sistema permaneça temporariamente inconsistente

Transações

Operações sobre o Banco de Dados consistente:

Inserção, atualização, etc.



Passos intermediários sobre o Banco de Dados:

Banco pode estar inconsistente



Banco de Dados consistente após todas as atualizações realizadas

Transações

| Cod_Pedido | Qtidade | Qtidade_Total |
|------------|---------|---------------|
| 001 | 100 | 850 |
| 002 | 250 | 850 |
| 003 | 500 | 850 |

Transações

| Cod_Pedido | Qtidade | Qtidade_Total |
|------------|---------|---------------|
| 001 | 100 | 850 |
| | | 850 |
| | | 850 |

$$\text{Qtidade_Total} = \sum \text{Qtidade}$$

$$850 = 100 + 250 + 500$$

Transações

| Cod_Pedido | Qtidade | Qtidade_Total |
|------------|---------|---------------|
| 001 | | |
| 002 | | |
| 003 | 500 | 850 |
| 004 | 200 | |

Inserção de novos dados

Transações

| Cod_Pedido | Qtidade | Qtidade_Total |
|------------|---------|---------------|
| 001 | 100 | 1050 |
| | | 1050 |
| | | 1050 |
| | | 1050 |

Atualização

$$\text{Qtidade_Total} = \sum \text{Qtidade}$$

$$1050 = 100 + 250 + 500 + 200$$

Transações

- O exemplo anterior envolveu 2 operações:
 - uma operação de INSERÇÃO de dados. Durante a realização desta operação os dados de Qtdade_Total de todos os pedidos ficaram inconsistentes (incorretos)
 - uma operação de ATUALIZAÇÃO para atualizar o valor de Qtdade_Total. Após a realização desta operação o banco de dados se tornou consistente, já que os valores do atributo Qtdade_Total foram atualizados

Transações

- Em uma transação pode ocorrer que uma das operações da transação não seja executada. Se isso acontece o banco de dados fica em estado inconsistente. Alguns exemplos:
 - poderia ocorrer uma queda de sistema entre as operações de INSERÇÃO e ATUALIZAÇÃO
 - poderia ocorrer um overflow aritmético na operação de ATUALIZAÇÃO

Transações

- **Gerenciamento de transações**

- um sistema que admite o gerenciamento de transações garante que se a transação executar algumas atualizações e ocorrer uma falha (por qualquer motivo) antes da transação atingir seu término planejado, então essas atualizações serão desfeitas

Transações

- Com o gerenciamento de transações, uma transação ou será executada integralmente ou será totalmente cancelada, isto é, será como se ela nunca tivesse executada
- Gerenciador de transações utiliza as operações COMMIT e ROLLBACK para manter um banco de dados consistente

Transações

- **Operação COMMIT**

- indica o término de uma transação bem-sucedida
- informa ao gerenciador de transações que uma unidade lógica de trabalho (transação) foi concluída com sucesso

Transações

- **Operação COMMIT (continuação)**
 - informa ao gerenciador de transações que o banco de dados está novamente em um estado consistente
 - informa ao gerenciador de transações que todas as atualizações feitas pela transação podem agora ser validadas ou tornadas permanentes no banco de dados

Transações

- **Operação ROLLBACK**

- indica o término de uma transação mal-sucedida
- informa ao gerenciador de transações que algo saiu errado durante a execução da transação

Transações

- **Operação ROLLBACK (continuação)**
 - informa ao gerenciador de transações que o banco de dados pode estar em estado inconsistente
 - informa ao gerenciador de transações que todas as atualizações feitas pela transação até agora devem ser retomadas ou desfeitas

Transações

| Cod_Pedido | Qtidade | Qtidade_Total |
|------------|---------|---------------|
| 001 | 100 | 850 |
| 002 | 250 | 850 |
| 003 | 500 | 850 |
| 004 | 200 | |

Transações

- Nesse exemplo emitiremos uma instrução COMMIT se tivermos passado pelas 2 atualizações com sucesso, o que acarretará a validação das 2 alterações e as tornarão permanentes no banco
- Se algo sair errado, ou seja, se uma das atualizações resultar em uma condição de erro, emitiremos uma instrução ROLLBACK

Transações

- **Como é possível desfazer uma atualização?**
 - através de um log de sistema (presente em fita ou disco), no qual são registrados detalhes de todas as operações de atualização
 - sendo assim, para desfazer alguma atualização, o sistema usará a entrada de log correspondente para restaurar o objeto atualizado a seu valor anterior

Transações

- Um COMMIT estabelece um ponto de COMMIT ou validação (também conhecido como ponto de sincronização em produtos comerciais)
- Um ponto de COMMIT corresponde, então, ao fim de uma unidade lógica de trabalho (transação) e, em consequência, a um ponto no qual o banco de dados deverá estar em um estado consistente

Transações

- ROLLBACK devolve o banco de dados ao estado em que ele se encontrava no ponto de COMMIT anterior
- COMMIT e ROLLBACK terminam a transação, não o programa. Em geral, a execução de um único programa consistirá em uma sequência de várias transações executadas uma após a outra

Transações

- Um ponto de COMMIT é estabelecido quando:
 - todas as atualizações feitas pelo programa em execução desde o ponto de COMMIT anterior são validadas
 - ou seja, quando as atualizações se tornam permanentes

Transações

- Antes do ponto de COMMIT:
 - todas as atualizações devem ser consideradas como apenas tentativas
 - ou seja, tentativas no sentido de que podem ser desfeitas subsequentemente (retomadas)
 - uma vez validada, uma atualização tem a garantia de que nunca será desfeita (essa é a definição de validação)

Transações

- Se uma transação for concluída com sucesso, então o sistema garantirá que suas atualizações serão instaladas permanentemente no banco de dados, mesmo que o sistema caia no momento seguinte

Transações

- Exemplo:
 - o sistema pode cair depois da instrução COMMIT ser aceita, mas antes das atualizações terem sido gravadas fisicamente no banco de dados, pois elas ainda podem estar esperando em um buffer de memória principal e serem perdidas no instante da queda

Transações

- Exemplo (continuação):
 - mesmo que isso aconteça, o procedimento de reinicialização do sistema ainda instalará essas atualizações no banco de dados, pois ele é capaz de descobrir os valores que devem ser gravados através do exame das entradas relevantes no log

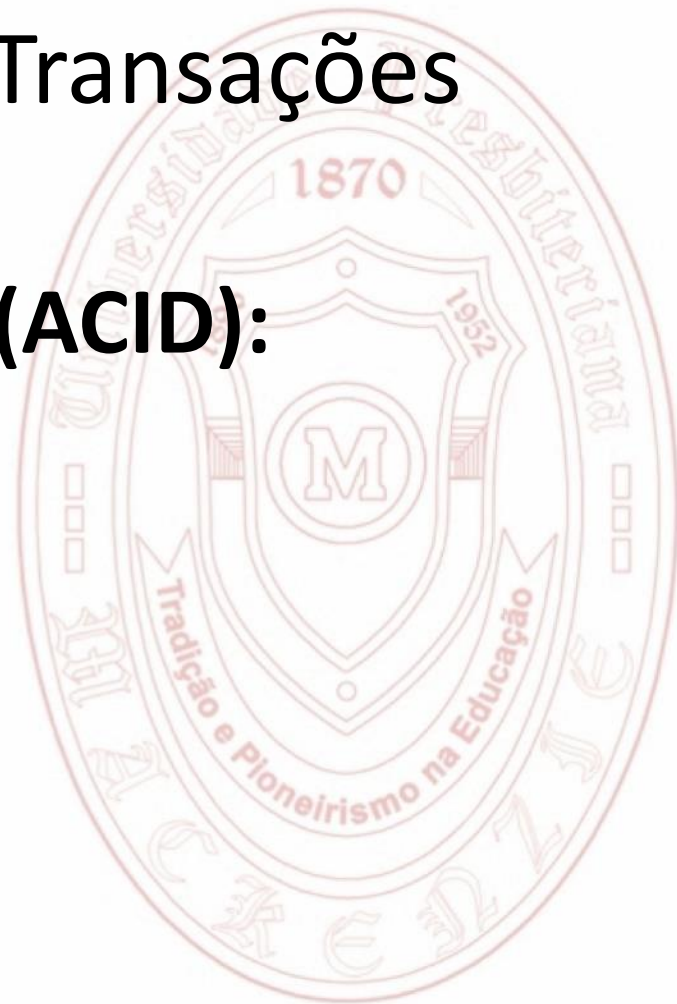
Transações

- Como o sistema pode consultar o log em caso de queda, então o **log deve ser fisicamente gravado antes de se completar o processamento de COMMIT**
- Essa é a **regra de gravação antes do registro no log**
- Assim, o procedimento de reinicialização recuperará qualquer transação concluída com sucesso que não tenha conseguido fazer com que suas atualizações fossem gravadas fisicamente antes de uma queda

Propriedades das Transações

Propriedades das transações (ACID):

- **A**tomicidade
- **C**onsistência
- **I**solamento
- **D**urabilidade



Propriedades das Transações

- **Atomicidade**

- as transações são atômicas (tudo ou nada)
- ou seja, uma transação ou será executada integralmente ou será totalmente cancelada

Propriedades das Transações

- **Consistência**

- as transações preservam a consistência do banco de dados
- ou seja, uma transação transforma um estado consistente do banco de dados em outro estado consistente, sem necessariamente preservar a consistência em todos os pontos intermediários da execução da transação

Propriedades das Transações

- **Consistência**

- após uma transação ter sido concluída, o banco de dados deve permanecer em um estado consistente, ou seja, deve satisfazer as condições de consistência e restrições de integridade previamente assumidas

Propriedades das Transações

- **Isolamento**

- as transações são isoladas umas das outras
- ou seja, embora haja muitas transações sendo executadas de modo concorrente, as atualizações de qualquer transação são ocultas de todas as outras até o commit dessa transação

Propriedades das Transações

- **Isolamento**

- se duas transações estão sendo executadas concorrentemente, seus efeitos devem ser isolados uma da outra
- esta propriedade está relacionada ao controle de concorrência do SGBDR

Propriedades das Transações

- **Isolamento (continuação)**

- exemplo: considere 2 transações distintas T1 e T2

- T1 só poderia ver as atualizações de T2 após T2 fazer o commit ou
 - T2 poderia ver as atualizações de T1 após T1 fazer o commit, mas certamente não ambas

Propriedades das Transações

- **Durabilidade**

– uma vez que a transação é concluída, suas atualizações sobrevivem no banco de dados mesmo que haja uma queda subsequente do sistema

Propriedades das Transações

- **Durabilidade**

- uma vez que uma transação ocorreu com sucesso, seu efeito não poderá mais ser desfeito, mesmo em caso de falha
- esta propriedade está relacionada à capacidade de recuperação de falhas do SGBD

Referência Bibliográfica

- SILBERSCHATZ, A., KORTH, H. F., SUDARSHAN, S. ***Sistema de Bancos de Dados***, 5 ed., Campus, 2006.

Obrigado

Profa. Elisângela Botelho Gracias
elisangela.botelho@mackenzie.br