

Capítulo 6

Transações

Motivação:

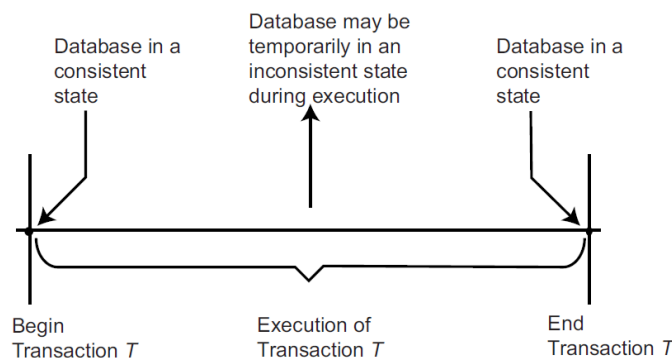
- Base de Dados: Obedece a todas as restrições de consistência (integridade) definidas sobre ele.
- Transação: Ações de transações concorrentes
- Base de Dados Distribuídos: Equivalência de uma única cópia: Uma base de dados replicada encontram-se num estado mutuamente consistente, se todas as cópias de todos os itens de dados que ele contém apresentam valores idênticos.

Transações

Definição:

Transação é uma execução de um programa de utilizador visto pelo SGBD como uma série de operações de leitura e escrita.

- É um programa em execução ou processo que inclui um ou mais acessos à base de dados que efetuam leituras ou atualizações dos seus registos.
- É uma unidade atômica de trabalho que estará completa ou não foi realizada.



O que uma transação faz?

- Toma posse sobre uma base de dados
- Executa uma ação sobre ela
- Gera uma nova versão da base de dados

Uma transação é vista pelo SGBD como uma série ou lista de ações:

- Leitura
- Escrita

Além de ler e escrever, uma transação deve especificar a ação final:

- Efetivação (*commit*)
- Cancelamento (*abort*)

Formalizando...

- $O_{ij}(x)$ alguma operação O_j da transação T_i que opera sobre uma entidade da base de dados x
- $O_{ij}(x) \in \{\text{leitura, escrita}\}$
- OS_i é o conjunto de todas as operações em $T_i \{OS_i = \cup_j O_{ij}\}$
- N_i é a condição de término de T_i . Onde $N_i \in \{\text{abortar, consolidar}\}$
- T_i é uma ordenação parcial sobre suas operações e a condição de término:

$$T_i = \{\Sigma_i, <i\}$$

1. $\Sigma_i = OS_i \cup \{N_i\}$
2. Para duas operações quaisquer $O_{ij}, O_{ik} \in OS_i$, se $O_{ij} = \{R(x) \text{ ou } W(x)\}$ e $O_{ik} = W(x)$ para qualquer item de dados x , então $O_{ij} <_i O_{ik}$ ou $O_{ik} <_i O_{ij}$
3. $\forall O_{ij} \in OS_i, O_{ij} <_i N_i$

Estados das Transações
<ul style="list-style-type: none"> • <u>Activado</u>: Durante a execução • <u>Parcialmente submetida</u>: Depois que a declaração final for submetida • <u>Falha</u>: após a descoberta de que a execução normal já não pode prosseguir • <u>Anulada</u>: após a reversão e a BD é restaurada ao seu estado inicial • <u>Submetida</u> - Conclusão bem sucedida

FLIGHT(FNO, DATE, SRC, DEST, STSOLD, CAP)
 CUST(CNAME, ADDR, BAL)
 FC(FNO, DATE, CNAME, SPECIAL)

```

Begin.transaction Reservation
begin
  input(flight_no, date, customer_name);
  EXEC SQL SELECT STSOLD, CAP
    INTO      temp1, temp2
    FROM      FLIGHT
    WHERE     FNO = flight_no AND DATE = date;
  if temp1 = temp2 then
    output("no free seats");
    Abort
  else
    EXEC SQL UPDATE FLIGHT
      SET      STSOLD = STSOLD + 1
      WHERE   FNO = flight_no AND DATE = date;
    EXEC SQL INSERT
      INTO     FC(FNO, DATE, CNAME, SPECIAL);
      VALUES (flight_no, date, customer_name, null);
    Commit
    output("reservation completed")
  endif
end.

```

EXEMPLO

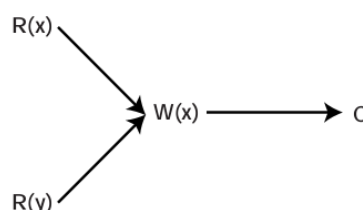
Considere a transação T simples

$\Sigma = \{R(x), R(y), W(x), C\}$
 $\prec = \{(R(x), W(x)), (R(y), W(x)), (W(x), C), (R(x), C), (R(y), C)\}$
 Read(x)
 Read(y)
 $x \leftarrow x + y$
 Write(x)
 Commit

A formalização é:

onde (O_i, O_j) como um elemento da relação $<$ que indica que $O_i < O_j$

A formalização é:



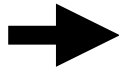
Simplificando...

- Objeto a ser lido: O
- Transação: T
- Leitura: $R_T(O)$
- Escrita: $W_T(O)$
- Efetivação: $Commit_T$
- Efetivação: $Abort_T$

Propriedades das Transações:

Acrônimo ACID

- Atomicidade
- Consistência
- Isolamento
- Durabilidade



Garantem os aspectos de consistência e confiabilidade das transações

Atomicidade

A propriedade de atomicidade garante que as transações sejam atômicas (indivisíveis). A transação será executada totalmente ou não será executada.

A atomicidade exige que, se a execução de uma transação for interrompida por qualquer tipo de falha, o SGBD seja responsável pela determinação do que fazer com a transação após a recuperação da falha.

- Tipos de falhas:
 - Cancelada ou terminada sem sucesso pelo SGBD por causa de alguma anomalia durante a execução
 - Sistema pode falhar (fonte de alimentação interrompida)
 - Transação pode encontrar uma situação inesperada (incapacidade de acessar um dado no disco)

Um SGBD garante a atomicidade de transação *desfazendo* as ações das transações incompletas

O SGBD mantém um registro, chamado Log, de todas as gravações feitas na Base de Dados.

Consistência

A propriedade de consistência garante que a base de dados passará de uma forma consistente para outra forma consistente.

- As transações transformam as BD consistentes em outras BD consistentes
- As transações não violam as restrições de integridade da BD

Grau 0	Transações T não substitui dados "sujeitos" de outras transações
Grau 1	Grau 0 + T não confirma nenhuma escrita antes de EOT
Grau 2	Grau 1 + T não lê dados "sujeitos" de outras transações
Grau 3	Grau 2 + Outras Transações não "suja" nenhuma informação lida por T antes de T estar completa

Isolamento

A propriedade de isolamento garante que a transação não será interferida por nenhuma outra transação concorrente.

Uma transação em andamento não pode revelar seus resultados a outras transações concorrentes antes de se consolidar

Durabilidade

A propriedade de durabilidade garante que o que foi salvo, não será mais perdido. Propriedade que assegura que, uma vez que a transação se consolida, seus resultados tornam-se permanentes e não podem ser apagados da base de dados.

Plano de Execução:

É uma lista de ações (leitura, escrita, cancelamento ou efetivação) de um conjunto de transações.

A ordem na qual duas ações de uma transação T aparecem num plano de execução deve ser a mesma em que elas aparecem em T.

T1	T2
R(A)	
W(A)	
	R(B)
	W(B)
R(C)	
W(C)	
COMMIT	COMMIT

Um plano de execução envolvendo duas transações

Plano de Execução Completo

Contém um cancelamento (*abort*) ou uma efetivação (*commit*) para cada transação cujas as ações estão listadas nele

Plano de Execução Serial

Transações são executadas no início ao fim uma por uma

Execução Concorrente:

O SGBD intercala as ações de diferentes transações para melhorar o desempenho:

- Enquanto uma transação está a espera que uma página seja lida do disco, o CPU pode processar outra transação
- Reduz o tempo que discos e processadores ficam inativos
- Aumenta o *throughput* do sistema
- Execução intercala entre uma transação curta e longa, permite que a curta termine mais rápido

Manutenção da consistência entre transações

- Se duas transações concorrentes acessam um item de dados que está a ser atualizado por uma delas, não é possível garantir que a segunda lerá o valor correto

T1	T2
R(x)	R(x)
$x <- x+1$	$x <- x+1$
W(x)	W(x)
COMMIT	COMMIT

$x = 50$

EXEMPLO

Considere as transações

T1: R(x)	T1: R(x)
T1: x <- x+1	T1: x <- x+1
T1: W(x)	T2: R(x)
T1: COMMIT	T1: W(x)
T2: R(x)	T2: x <- x+1
T2: x <- x+1	T2: W(x)
T2: W(x)	T1: COMMIT
T2: COMMIT	T2: COMMIT
RESULTADO	RESULTADO
T1 = 51	T1 = 51
T2 = 52	T2 = 51

Anomalias em Razão da Execução Concorrente

- Dados sujos
 - T2 lê um objeto de dados gravado anteriormente por T1 que ainda não foi efetivado (WR)
- Leitura não repetível ou vaga
 - Uma transação T2 grava um valor lido, anteriormente, por T1 ainda em andamento (RW)
- Sobrescrevendo dados não efetivados
 - Uma transação T2 pode sobrescrever o valor de um objeto X o qual já foi modificado por T1, enquanto T1 ainda está em andamento (WW)

Tipos de Transações:

As transações classificam-se de acordo com diversos critérios:

- Duração da Transação

- On-line ou curtas

- Tempos de execução/resposta muito curtos e pelo acesso a uma porção relativamente pequena da base de dados

- Em lote ou longas

- Tempos de execução/resposta longos e acessam uma grande porção da base de dados

- Estrutura

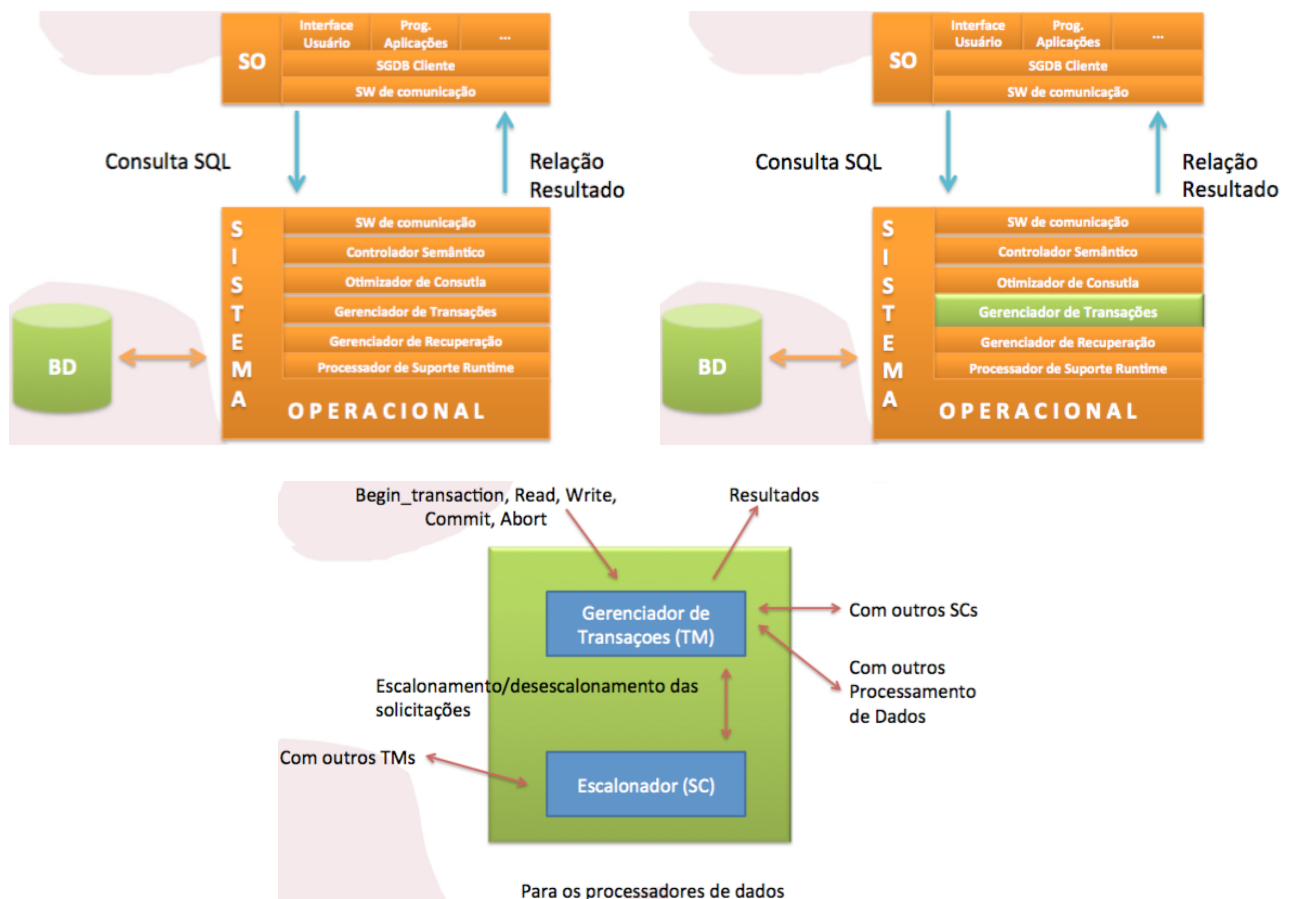
- Planas

- Tem um único ponto de início e um único ponto de fim

- Aninhadas

- Permite a inclusão de outras transações com seus próprios pontos de início e fim

Revisão da Arquitetura:



- **Gestor de Transações (TM)**
 - Responsável pela coordenação da execução das operações da base de dados em nome de uma aplicação
- **Escalonador (SC)**
 - Responsável pela implementação de um algoritmo específico de controle de concorrência para sincronizar o acesso à Base de Dados.
- **Conjunto de Gestores de Recuperações Locais**
 - Responsável por implementar os procedimentos locais pelos quais a Base de Dados pode ser levado a um estado consistente depois de uma falha.

Comandos:

O gestor de transação fornece cinco comandos:

- **Begin_Transaction**
 - Indica para o TM que uma nova transação está começando. O TM regista o nome da transação, a aplicação de origem, entre outras atividades.
- **Read**
 - Se o item de dados x estiver armazenado no local, seu valor será lido e retornado para a transação. Caso contrário, o TM selecionará uma cópia de x e solicitará que sua cópia seja devolvida.
- **Write**
 - O TM coordena a atualização do valor de x em cada site em que ele reside.
- **Commit**
 - O TM coordena a atualização física de todas as Base de Dados que contém cópias de cada item de dado para o qual foi emitido um comando de gravação.
- **Abord**
 - O TM certifica-se de que nenhum efeito da transação se refletiu na Base de Dados.

Com esses serviços, o TM pode comunicar-se com os SCs e com os processadores de dados no mesmo site ou em sites diferentes.