INSTITUTO FEDERAL
Rio Grande do Norte | Campus Parnamirim

Gerenciamento de Transações

Prof. Fábio Procópio

fabio.procopio@ifrn.edu.br

Tecnologia em Sistemas para Internet – Programação em Bancos de Dados

Relembrando...

- Na aula passada, apresentamos uma estratégica bastante útil utilizado na programação de banco de dados: *stored procedures*;

- O conteúdo desta aula tem uma abordagem mais teórica do que, em geral, aplicamos aqui. No entanto, apresentaremos alguns exemplos práticos do que se será discutido. Vamos nessa?!

Introdução

- Transação é uma unidade lógica de trabalho que acessa e, possivelmente, atualiza vários itens de dados
  - Ela é executada integralmente ou totalmente cancelada
- Uma transação é
  - iniciada com a operação **BEGIN TRANSACTION**
  - encerrada com a operação **COMMIT** (com sucesso) ou **ROLLBACK** (com falha);
- No escopo de uma transação, podem existir diversas operações
  - Geralmente, ocorre uma sequência de várias operações de inserção, alteração ou exclusão organizadas em blocos atômicos
- Uma transação tem como objetivo transformar um estado correto do banco de dados em um outro estado também correto.

Exemplo

- Imagine uma transação que consiste em uma transferência bancária no valor de R$ 100,00 entre as contas correntes 123456 e 654321...

- O pseudocódigo da transação bancária poderia ser escrito como segue:

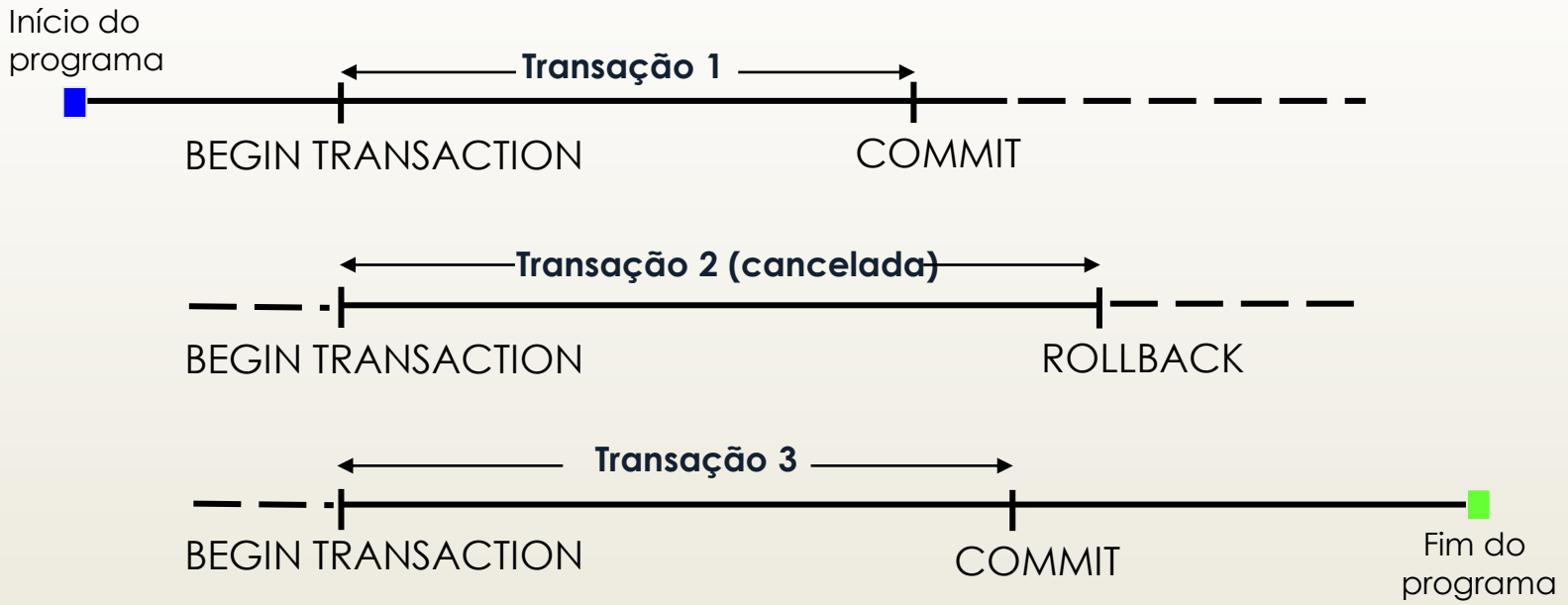
```
BEGIN TRANSACTION
   UPDATE TbConta
      SET VaSaldo = VaSaldo - 100
    WHERE NuConta = 123456

   UPDATE TbConta
      SET VaSaldo = VaSaldo + 100
    WHERE NuConta = 654321

    IF nenhum erro foi identificado
      COMMIT TRANSACTION
    ELSE
      ROLLBACK TRANSACTION
```

Gerenciamento de transações

- Um programa é executado com base em uma sequência de transações



- O **gerenciador de transações** é um componente do SGBD que fornece o conceito de **atomicidade** à transação
  - O seu funcionamento se dá sobre as operações de **COMMIT** e **ROLLBACK**

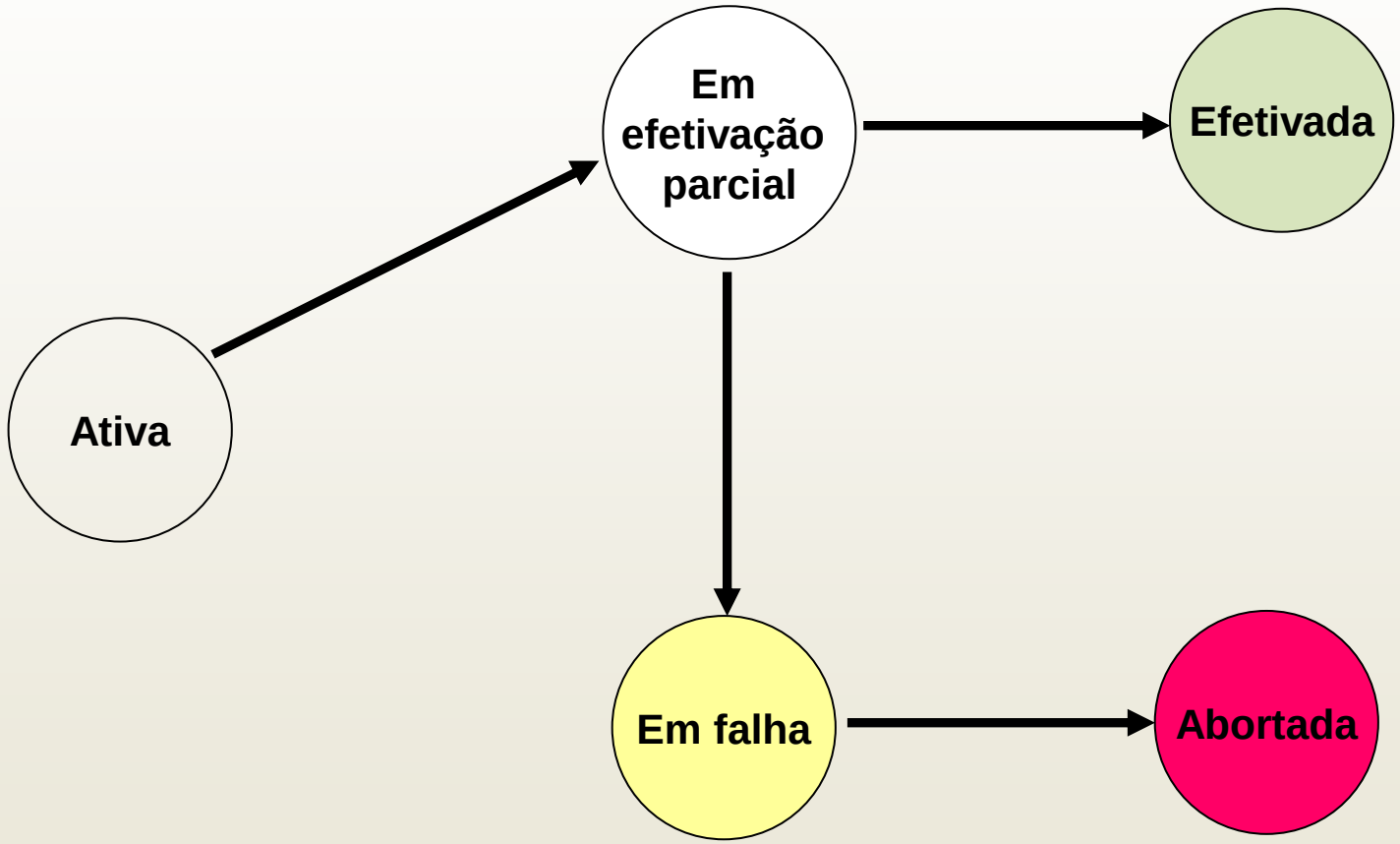
COMMIT e ROLLBACK

- Operação **COMMIT**
  - Indica o **término bem sucedido** de uma transação
  - Informa ao gerenciador de transações que
    - A transação foi concluída com sucesso
    - O BD se encontra em um estado correto
    - Todas as alterações realizadas pela transação devem ser confirmadas no BD
- Operação **ROLLBACK**
  - Assinala o **término mal sucedido** de uma transação
  - Informa ao gerenciador de transações que
    - Algo saiu errado
    - O BD pode estar em um estado incorreto
    - Todas as alterações realizadas devem ser desfeitas

Propriedades ACID

- A integridade dos dados é garantida por meio de 4 importantes propriedades (ACID)
  - **A**tomicidade (*Atomicity*)
    - Garante que as transações sejam indivisíveis
    - Uma transação é executada **integralmente** ou **totalmente cancelada**
  - **C**onsistência (*Consistency*)
    - Garante que o BD passará de um estado consistente para outro consistente
  - **I**solamento (*Isolation*)
    - Garante que uma transação não sofrerá interferência de uma outra concorrente
    - Uma transação não toma conhecimento de outras transações concorrentes
  - **D**urabilidade (*Durability*)
    - Depois de uma transação ser concluída com sucesso, é garantido que, mesmo havendo falhas no sistema, os dados serão efetivados no BD

Estados de uma transação – 1 de 2



Estados de uma transação – 2 de 2

- **Ativa**
  - É o estado inicial
  - Permanece nesse estado enquanto estiver em execução
- **Em Efetivação Parcial**
  - Após execução da última declaração
- **Efetivada**
  - Após a conclusão bem sucedida
- **Em Falha**
  - Após descoberta de que a execução normal não pode ser realizada
- **Abortada**
  - Após a transação ser desfeita e o BD ser restabelecido ao estado anterior do início da execução da transação

Falhas

- Um SGBD deve estar preparado para se recuperar não apenas de **falhas locais** (por exemplo, *overflow*), mas também de **falhas globais** como uma queda de energia:
  - Uma falha local afeta apenas a transação em que a falha realmente ocorreu
  - Já a global implica em todas as transações que estão em andamento no instante da falha e isso implica em todo o sistema

- Essas falhas se enquadram em categorias maiores
  - **Falhas de sistema** (*soft crash*): queda de energia, mau funcionamento do hardware, bug no SGBD ou no SO
  - **Falhas de mídia** (*hard crash*): quebra do cabeçote de leitura/escrita, falha durante a transferência de dados
  - **Falhas de transação**: erro lógico (entrada inadequada, dado não encontrado, *overflow* ou *underflow*, limite de recurso excedido) e erro de sistema (*deadlock*)

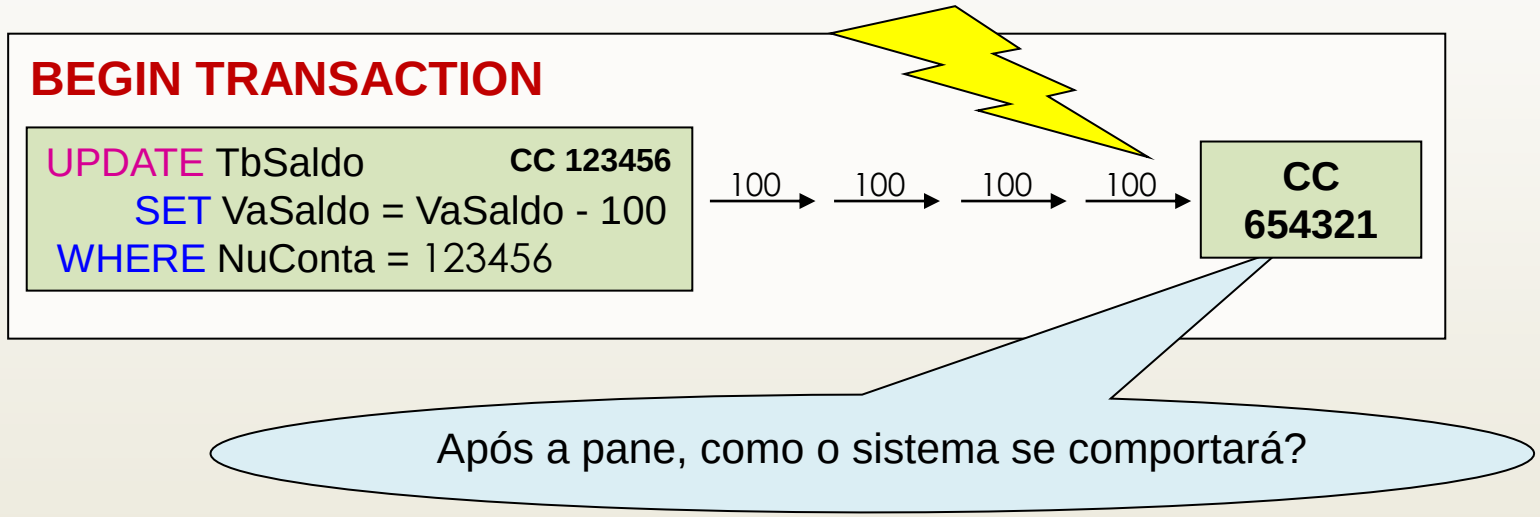
Recuperação de falhas – 1 de 2

- Consiste em um processo que garante o retorno do banco de dados ao seu estado consistente mais recente, antes da ocorrência da falha;

- As possíveis falhas são categorizadas em:
  - Falha catastrófica
    - Restaura o *backup* mais recente
    - Reconstrói um estado consistente do BD refazendo as operações já confirmadas a partir do arquivo de log
  - Falha não-catastrófica
    - Reverte as alterações que causaram a inconsistência
    - Desfaz (ou refaz) as operações ainda não confirmadas no BD
- As principais técnicas para recuperação de falhas não-catastróficas utilizam arquivos de log e são conhecidas como: atualização postergada e atualização imediata

Recuperação de falhas – 2 de 2

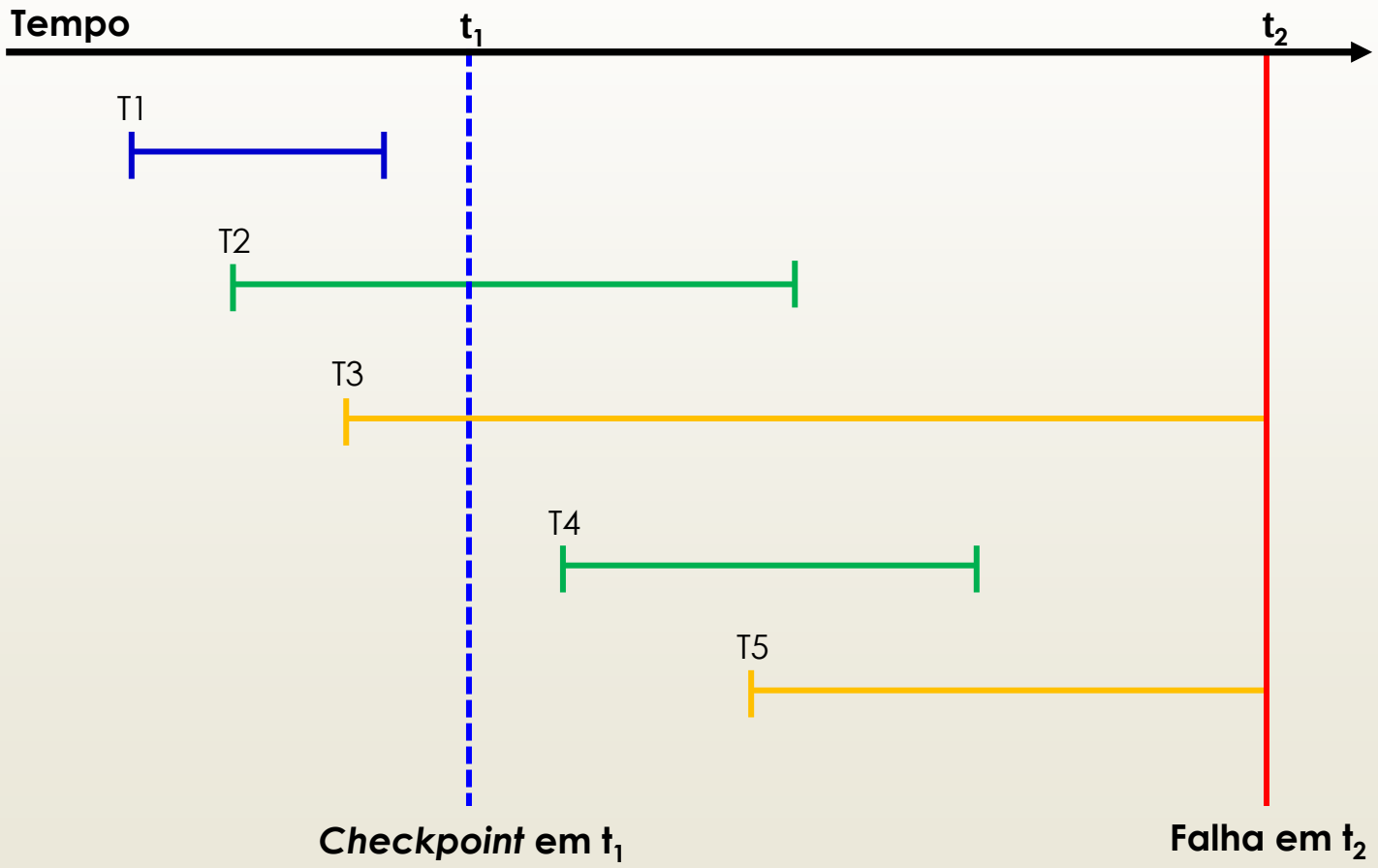
- Imagine a seguinte situação: um correntista inicia uma transferência de R$ 100,00. No meio do procedimento, um raio cai sobre a rede elétrica interrompendo o serviço de energia do banco.



Checkpoint

- Mecanismo utilizado para **identificar o estado de um BD** em um determinado instante T;
- A marcação de um *checkpoint* grava, em disco
  - o conteúdo dos *buffers* de alterações
  - um registro do *checkpoint* no log físico
- Um registro de *checkpoint* lista todas as **transações** que estavam **em andamento** quando o ponto de checagem foi marcado;
- As **modificações** nas páginas de um banco de dados são **executadas em memória** e não são, necessariamente, gravadas no disco após cada atualização;
- Periodicamente, o SGBD deve **executar** os registros de ***checkpoint* para aplicar as modificações** (que estão em memória) em disco.

Cenário de uma falha – 1 de 2

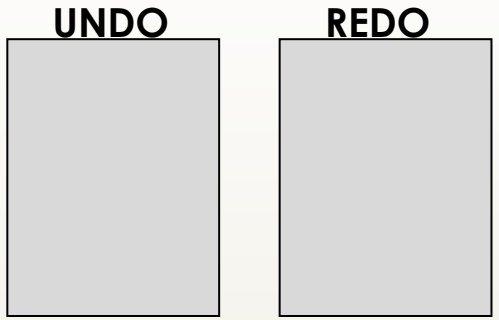


Cenário de uma falha – 2 de 2

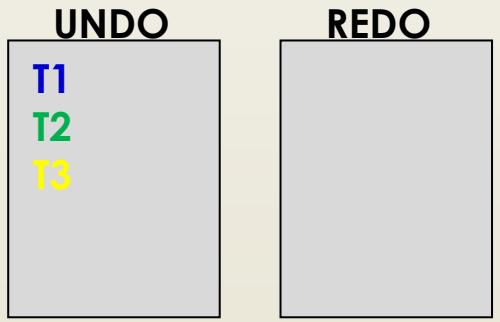
- A transação T1 foi finalizada com sucesso antes da execução do checkpoint (instante t1);
- T2 foi iniciada antes do instante t1 e foi finalizada depois de t1 e antes de t2 (ocorrência da falha);
- T3 começou antes do instante t1, mas ainda não havia sido finalizada quando ocorreu a falha em t2;
- T4 começou depois do instante t1 e foi finalizada com sucesso antes da ocorrência da falha (t2);
- Finalmente, T5 também começou depois do instante t1 mas não foi finalizada antes da ocorrência da falha (t2).

Algoritmo de recuperação – 1 de 3

1. Cria duas listas: UNDO e REDO



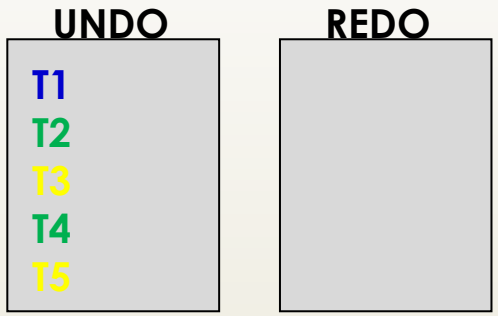
2. Em UNDO, colocar todas as transações iniciadas antes do último *checkpoint* e esvazia REDO (caso existam transações)



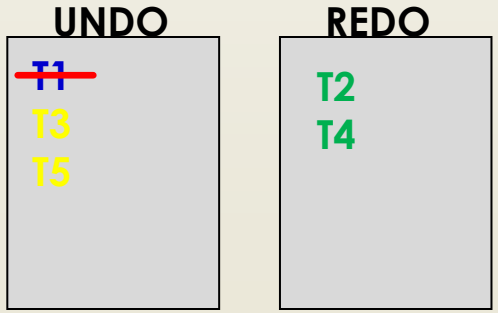
Algoritmo de recuperação – 2 de 3

3. A partir do último *checkpoint*, pesquisar o log para a frente

4. Se for encontrado no log um registro de BEGIN TRANSACTION para a transação T, adicionar T à lista UNDO



5. Se um COMMIT de uma transação T for encontrada no log, mover T de UNDO para REDO



Algoritmo de recuperação – 3 de 3

6. O sistema percorre o log do fim para o início desfazendo as transações da lista UNDO: T3 e T5;

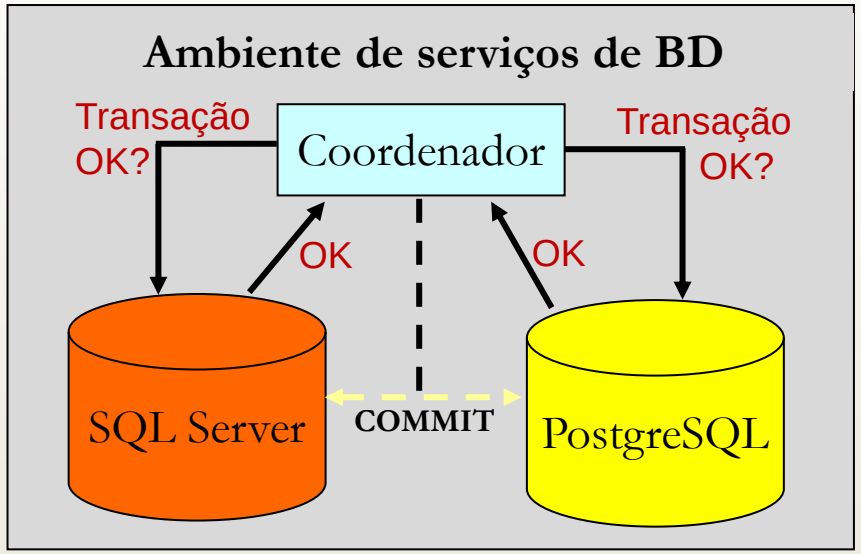
7. Em seguida, percorre o log para a frente refazendo as transações da lista REDO: T2 e T4.

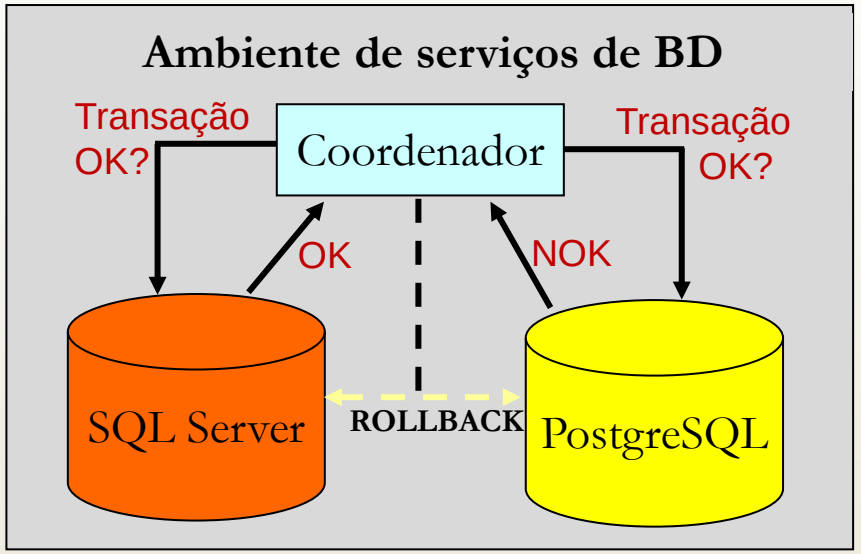
Nota:

A restauração do banco de dados a um estado correto, refazendo o trabalho, pode ser chamada de **recuperação direta.**

Já a restauração do banco de dados a um estado correto, desfazendo o trabalho, pode ser chamada de **recuperação inversa**.

COMMIT de Duas Fases





Referências

1. DATE, C J. **Introdução a Sistemas de Banco de Dados**. Rio de Janeiro: Elsevier, 2003.

2. SILBERSCHATZ, Abraham, KORTH, Henry F. e SUDARSHAN, S. **Sistema de Banco de Dados**. São Paulo: MAKRON Books, 1999.

3. TUTORIALS POINT. **DBMS - Transaction.** Disponível em: https://www.tutorialspoint.com/dbms/dbms_transaction.htm. Acessado em: 06 out 2019.

4. DEVMEDIA. **Conceitos e Criação da View – Views no SQL Server – Parte 1.** Disponível em: http://www.devmedia.com.br/conceitos-e-criacao-da-view-views-no-sql-server-parte-1/22390. Acessado em: 23 jul 2019.

5. SQL PARA TODOS. **Modelo ACID SQL**. Disponível em: http://sqlparatodos.com.br/modelo-acid-sql/. Acessado em: 06 out 2019.'