

Disciplina: *Sistemas de Banco de Dados 1*

Turma: A - 2º/2019

Professor: *Vandor Roberto Villardi Rissoli*

Alunos: *Rogério S. dos Santos Júnior*

Matrícula: 17/0021751

*Iolane C. A. de Andrade*

Matrícula: 13/0028355

## **AULA 12 – RECUPERAÇÃO DE BANCO DE DADOS**

### ***1) Existem falhas no Banco de Dados?***

As falhas são classificadas como falhas de transação, sistema e mídia. Existem alguns motivos possível para uma transação falhar: Falha do computador, Falha de Transação ou do sistema, Erros locais ou condições detectadas pela transação; Imposição de controle de concorrência, Fala de disco; Problema físico e catástrofes.

**Falha do computador (falha do sistema):** É um erro de hardware, software ou rede que ocorre durante a execução da transação.

**Erro de transação ou erro de sistema:** A falha de transação pode ocorrer devido a valores de parâmetro errôneos ou erro lógico de programação. Exemplo: estouro de inteiro ou divisão por zero.

**Erros locais ou condições de execução detectadas pela transação:** Ocorre quando, durante a execução da transação condições de cancelamento necessitam ser acionadas. Exemplo: quando dados da transação não podem ser encontrados, como saldo de conta insuficiente, isso pode levar a uma transação de saque não ser executada.

**Imposição de controle de concorrência:** Ocorre quando o método de controle de concorrência decide abortar uma transação por estar violando a serialização. Ou pode abortar uma ou mais transações para resolver um estado de deadlock. Geralmente elas são reiniciadas automaticamente posteriormente.

**Falha de disco:** Alguns blocos de disco podem perder os dados devido a defeitos de leitura, gravação, ou defeito na cabeça de leitura/gravação. Isso pode ocorrer durante a leitura/gravação de uma transação.

**Problemas físicos e catástrofes:** Isso se refere a diversos problemas como, falhas de energia ou de ar-condicionado, incêndio, roubo, sabotagem, regravação de disco ou fitas por engano e montagem da fita errada pelo operador.

## ***2) Existe como recuperar as falhas no Banco de Dados?***

A recuperação de falhas de transação em geral significa que o banco de dados é restaurado ao estado consistente mais recente, antes do momento da falha. Essa informação costuma ser mantida no *log do sistema*. Estratégias típicas de recuperação podem ser resumidas em:

- Se houver um dano extensivo a uma grande parte do banco de dados devido a uma falha catastrófica, como falha de disco, o método de recuperação é restaurar uma cópia antiga de *backup* para o arquivamento e reconstruir um estado mais recente, replicando ou refazendo as operações das transações confirmadas do log em *backup* até o momento da falha.
- Quando a falha não foi catastrófica, a estratégia é recuperação e identificação de qualquer mudança que possa ter gerado uma inconsistência no banco de dados. Exemplo: transação que atualizou itens do banco de dados, mas não registrou necessidade de ter suas mudanças revertidas ao desfazer suas operações. Também pode ser preciso refazer algumas operações a fim de restaurar um estado consistente. Uma falha não catastrófica, não necessita de uma cópia de arquivamento completa do banco de dados, e sim das entradas registradas no log do sistema.

---

### ***3) O que significa as siglas relacionadas abaixo?***

#### **a) Log em Banco de dados**

Log de dados é um arquivo de texto gerado por um software para descrever eventos sobre o seu funcionamento, utilização por usuários ou interação com outros sistemas. Um log, após ser gerado, passa a ser incrementado ao longo do tempo com informações que permitem diagnosticar anormalidades em relação ao propósito do sistema e questões de segurança e acessibilidade.

Cada unidade de informação gravada no log contém basicamente o horário em que foi inserida, uma identificação do agente que motivou sua escrita e uma informação a respeito de uma modificação no estado de um sistema ou hardware.

Todo banco de dados do SQL Server tem um log de transações que registra todas as transações e as modificações de banco de dados feitas por cada transação.

#### **b) UNDO**

Desfaz as transações que não foram parcialmente efetivadas (o *start* e o *commit* não foram escritos no log). O processo de desfazer as operações *write\_item* de uma ou mais transações do log deve acontecer na ordem inversa da ordem em que as operações foram gravadas no log.

Se uma transação falhar depois de registrar mudanças no banco de dados, mas antes de alcançar seu estado de efetivação, os efeitos de suas operações no banco deverão ser desfeitos (UNDO).

#### **c) REDO**

Refaz as transações que foram parcialmente efetivadas (o *start* e o *commit* foram escritos no log). Examina as entradas de log e ajusta o valor do item X no banco de dados para novo valor (imagem posterior).

O REDO (refazer) será necessário no caso de o sistema falhar depois que a transação for efetivada, mas antes de todas as suas alterações serem gravadas em disco, utilizando o log.

#### d) Checkpoint em Banco de dados

O checkpoint grava as informações no *transaction log* para saber quais arquivos foram e não foram gravados no banco de dados. Após isso o checkpoint fica verificando periodicamente as transações que foram completadas com sucesso no transaction log, as quais ainda não foram gravadas no arquivo de dados do disco.

Caso acontecer alguma falha no banco de dados ou em outro módulo que vá afetar o funcionamento do banco de dados, um processo de recuperação automatizado é disparado na próxima vez que o banco de dados for inicializado, assim fazendo com que as transações completadas e que ainda não foram aplicadas no banco de dados sejam aplicadas nesse momento, assim garantindo a “Durabilidade” na transação.

Ações seguidas em um processo de *checkpoint* são:

1. Suspende temporariamente a execução de transações.
2. Gravar à força todos os buffers de memória que tenham sido modificados para o disco.
3. Gravar um registro [*checkpoint*] no log, forçando a gravação do mesmo no disco.
4. Retomar as transações em execução.

Para não atrasar o processamento de transações em função da ação 1 do processo de checkpoint, a técnica *fuzzy checkpoint* permite a retomada do processamento antes da conclusão da ação 2. Para isso, a técnica grava o registro de *checkpoint*, mas mantém um apontador para o checkpoint anterior (checkpoint válido) até que a ação 2 seja concluída.



---

## REFERÊNCIAS

ASSIS, Guilherme Tavares de. **Recuperação de Falhas**: banco de dados II. Disponível em: <  
[http://www.decom.ufop.br/guilherme/BCC441/geral/bd2\\_recuperacao-de-falhas.pdf](http://www.decom.ufop.br/guilherme/BCC441/geral/bd2_recuperacao-de-falhas.pdf) >.  
Acesso em: 25 nov. 2019.

MERCHSMANN, Luiz Henrique de Campos. **Noções de Recuperação de Falhas**. Disponível  
em: [http://www.decom.ufop.br/luiz/site\\_media/uploads/arquivos/bcc321/slides12\\_bdi.pdf](http://www.decom.ufop.br/luiz/site_media/uploads/arquivos/bcc321/slides12_bdi.pdf) .  
Acesso em: 25 nov. 2019.

UNIOESTE. **Técnicas de Recuperação**. Disponível em: <  
[http://www.inf.unioeste.br/~clodis/BDII/BDII\\_Modulo\\_3.pdf](http://www.inf.unioeste.br/~clodis/BDII/BDII_Modulo_3.pdf) >. Acesso em: 25 nov. 2019.

ELMARI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6ª Edição. Editora: Pearson. 2010.