Possíveis falhas em um banco de dados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Descrição** | **Causas** | **O que compromete** | **Probabilidade de ocorrência** |
| Falha de transação | De acordo com Date (2004), ao fazer uma transação é possível que ocorram erros durante seu processo, de modo a comprometer seu resultado. | Date (2004) cita como possíveis causas a queda do sistema entre operações ou um estouro aritmético, por exemplo. | Possíveis falhas durante um processo de transação deixam o estado do banco de dados incorreto, conforme é afirmado por Date (2004). O autor complementa que é necessária uma garantia de que as atualizações sejam executadas por completo e corretamente. | “Para nós, o ideal seria ter uma garantia sólida de que ambas as atualizações serão executadas. Infelizmente, é impossível fornecer tal garantia – há sempre a possibilidade de que as coisas saiam erradas, e saiam erradas no pior momento possível.” (DATE, 2004) |
| Falha de sistema | Uma falha global pode se classificar como uma falha de sistema “que afetam todas as transações em curso no momento, mas não danificam fisicamente o banco de dados. Às vezes, uma falha do sistema é chamada soft crash.” (DATE, 2004) | Date (2004) cita como um principal exemplo da causa de uma falha de sistema a queda de energia, o que acaba prejudicando todas as transações de um único banco de dados. | De acordo com Date (2004), a falha global afeta todas as transações em andamento e dessa forma prejudica o sistema como um todo. Além disso, o autor cita que quando falhas do sistema ocorrem é perdido o conteúdo da memória principal, de forma que o estado da transação se torna desconhecido. | Segundo Elmasri e Navathe, as falhas de sistema são comuns, portanto têm uma probabilidade relativamente grande de acontecer, de forma que “o sistema precisa manter informações suficientes para recuperar-se rapidamente da falha”. |
| Falha de mídia | “[...] uma falha de mídia é uma falha – como a queda de uma cabeça de disco, ou então uma falha do controlador de disco – na qual uma parte do banco de dados é destruída fisicamente.” (DATE, 2004). Date relembra também que as falhas de mídia são por vezes chamadas de *hard crash.* | Conforme descrito por Date (2004), as falhas de mídia são causadas por danos físicos ao banco de dados, tais como queda de uma cabeça de disco ou falha do controlador de disco. | As falhas de mídia “causam danos ao banco de dados ou a uma parte dele, e afetam pelo menos todas as transações que no momento estão usando essa parte” (DATE, 2004) | De acordo com o site platinumdatarecovery (c2021), a falha de mídia é um dos tipos de falha mais perigosos e, se ocorrer, o backup deve ser feito a tempo. É citado que a recuperação da falha de mídia leva mais tempo que qualquer outro tipo de falha. Além disso, de acordo com o site da IBM (c2021), esse tipo de falha pode ser difícil de se detectar e prevenir. |
| Imposição de controle de concorrência | “Controle de concorrência é quando, em um banco de dados, usuários distintos tentam acessar a mesma informação e então é feito um controle entre essas transações” (GUIRRA, 2013). | De acordo com Elmasri e Navathe (2011), ao ser realizado um controle de concorrência é possível que uma transação seja abortada por violar a serialização ou para solucionar um problema de deadlock. | Conforme é afirmado por Elmasri e Navathe (2011), a imposição de controle de concorrência pode ocasionar o aborto de alguma transação, entretanto elas costumam ser reiniciadas de modo automático posteriormente. | Falhas ocasionadas pela imposição de controle de concorrência podem ocorrer com frequência, conforme é afirmado por Elmasri e Navathe (2011). Os autores citam que esse é um tipo de falha comum de se acontecer e que o sistema deve guardar informações para se recuperar rapidamente. |
| Problemas físicos e catástrofes | Segundo Elmasri e Navathe (2011) esse tipo de problema envolve inúmeras possibilidades, tais como catástrofes, que de alguma forma podem prejudicar o banco de dados fisicamente. | Dentre os exemplos de imprevistos que podem prejudicar o banco de dados Elmasri e Navathe (2011) citam “falha de energia ou de ar-condicionado, incêndio, roubo, sabotagem, regravação de discos ou fitas por engano e montagem da fita errada pelo operador.” | Conforme Elmasri e Navathe (2011), esse tipo de problema “se refere a uma lista sem fim de problemas” que, portanto, podem afetar o banco de dados de maneiras distintas, o que torna sua recuperação mais difícil e menos específica. | De acordo com Elmasri e Navathe (2011), esse é um tipo de problema que, apesar de complicado, é raro e imprevisível. |

Técnicas de recuperação para possíveis falhas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Descrição** | **Funcionamento** | **Vantagens** | **Desvantagens** |
| Atualização Postergada | Segundo DEBATIN (2021), essa técnica é para recuperação de falhas não catastróficas. Essa técnica adia atualizações no BD até a transação completar sua execução e efetuar o *Commit*. | Segundo DEBATIN (2021), essa técnica funciona assim: As atualizações da transação são registradas somente no log e nos buffers do cache. Se a transação falhar antes do *commit*, não é necessário desfazer nenhuma operação. Depois que a transação é *comitada*, o log com as atualizações da transação é usado para gravar as atualizações no BD. | Ignora operações que estavam sendo executadas no momento da falha. Não precisa desfazê-las e refazê-las (DEBATIN, 2021). | Para grandes transações, necessita de um tamanho de cache compatível (DEBATIN, 2021). |
| Atualização Imediata | De acordo com DEBATIN (2021), essa técnica também é para recuperação de falhas não catastróficas, e atualiza o BD imediatamente, quando a transação emite um comando de atualização, sem ter que esperar o *Commit*. Se a transação não obter sucesso, é só refazê-la. | Segundo DEBATIN (2021), essa técnica mantém duas listas de transações (transações *comitadas* desde o último *checkpoint*, e transações ativas). Todas as operações de escrita de transações ativas são desfeitas, a partir do log, na ordem inversa em que foram gravadas, utilizando o *UNDO*. Todas as operações de escrita das transações *comitadas* são refeitas, a partir do log, na ordem em que foram gravadas, utilizando o *REDO*. O *UNDO* ajusta o valor do item de dado no BD para o valor antigo, e o *REDO* ajusta para o novo valor. | As operações são salvas imediatamente no BD;  Não há necessidade de refazer operações se a técnica de recuperação garantir que todas as atualizações de uma transação sejam registradas no banco de dados em disco antes do commit da transação. (DEBATIN, 2021). | Se a operação for *comitada* antes que todas as alterações sejam salvas no BD, é necessário refazê-la. |
| RDU-M | Segundo Elmasri e Navathe (2011), essa técnica é baseada na técnica de recuperação postergada, e é para sistemas multiusuários. | De acordo com Elmasri e Navathe (2011), essa técnica possui duas listas de transações (*Comitadas* desde o último *checkpoint*, e ativas). Basicamente, essa técnica refaz todas as operações de escrita das transações *comitadas* com base no arquivo de log (na ordem em que foram escritas), e cancela as transações ativas não *comitadas*, e as refaz depois. | As operações nunca precisam ser desfeitas, por causa que uma transação não registra quaisquer mudanças no BD em disco até que atinja seu *commit*, e por causa que uma transação nunca vai ler o valor de um item que é gravado por uma transação não *comitada* (ELMASRI; NAVATHE, 2011). | Limita a execução concorrente das transações, porque todos os itens bloqueados para a gravação permanecem bloqueados até que a transação atinja seu ponto de confirmação. Além disso, pode ser exigido um espaço de buffer excessivo para manter todos os itens atualizados até que as transações sejam *comitadas* (ELMASRI; NAVATHE, 2011). |
| Paginação de sombra | Segundo Elmasri e Navathe (2011), essa técnica serve tanto para sistemas monousuário quanto para sistemas multiusuários, e basicamente, é uma técnica que considera o BD uma série de páginas de disco de tamanho fixo. Essas páginas possuem diretórios que são mantidos na memória principal — se não for muito grande — e todas as leituras e gravações do BD passam por ela. Quando uma transação começa a ser executada, o diretório atual — cujas entradas apontam para as páginas de banco de dados mais recentes no disco — é copiado para um diretório de sombra. O diretório de sombra é, então, salvo em disco, enquanto o diretório ativo é usado pela transação. Durante a execução da transação, o diretório de sombra nunca é modificado. | Segundo Elmasri e Navathe (2011), quando uma operação de escrita é realizada, uma nova cópia da página de BD é feita, mas a cópia antiga dessa página não é modificada — é gravada em outro lugar (bloco de disco ainda não usado) —. A entrada do diretório atual é modificada para apontar para o novo bloco de disco, enquanto o diretório de sombra não é modificado e continua a apontar para o antigo bloco de disco não modificado. Para recuperar-se de uma falha durante a execução da transação, basta liberar as páginas de BD modificadas e descartar o diretório antigo. A confirmação de uma transação corresponde a descartar o diretório de sombra anterior. Como a recuperação não envolve desfazer nem refazer itens de dados, essa técnica pode ser categorizada como *NO-UNDO/ NO-REDO*. | Não usar log em ambientes monousuário (ELMASRI; NAVATHE, 2011). | * As páginas de BD atualizadas mudam de local no disco, tornando difícil manter próximas as que são relacionadas, sem o uso de complexas estratégias de gerenciamento e armazenamento (ELMASRI; NAVATHE, 2011).   Se o diretório for grande, há um overhead significativo, à medida que as transações são confirmadas (ELMASRI; NAVATHE, 2011). |
| Backup do BD | De acordo com Elmasri e Navathe (2011), essa técnica é para falhas catastróficas e, basicamente, é o fato de fazer periodicamente um backup do BD inteiro e do log em fitas magnéticas ou outros dispositivos de armazenamento offline de grande capacidade. | Segundo Elmasri e Navathe (2011), o backup é em fitas magnéticas ou outros dispositivos de armazenamento offline de grande capacidade. Esses dispositivos (fitas e outros) são armazenados em locais seguros como câmaras de armazenamento subterrâneas. Para evitar perder todos os efeitos das transações que foram executadas desde o último backup, é comum fazer o backup do log do sistema em intervalos mais frequentes do que o do banco de dados inteiro, copiando-o periodicamente para fita magnética. O log do sistema costuma ser muito menor do que o próprio banco de dados, e, portanto, pode ser copiado com mais frequência. Portanto, os usuários não perdem todas as transações que realizaram desde o último backup do banco de dados. Todas as transações *comitadas* e registradas no log que foi copiado na fita podem ter efeito sobre o BD refeito. Um novo log é iniciado após cada backup do banco de dados. Assim, para recuperar-se da falha do disco, o banco de dados é primeiro recriado no disco com base em sua cópia de backup mais recente em fita. Depois disso, os efeitos de todas as transações confirmadas, cujas operações foram registradas nas cópias do log do sistema, são refeitos. | O autor não cita, entretanto, acredito que seja o fato de não perder os dados mesmo em situações extremas (catástrofes) | O autor não cita, entretanto, acredito que seja o fato desses dispositivos (fitas e outros) estragarem e a mesmo serem roubados. |

REFERÊNCIAS:

DATE, Christopher J. **Introdução a Sistemas de Banco de Dados**. 8. ed. [*S. l.*]: Elsevier, 2004. 1623 p. ISBN 978-85-352-8445-4.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011. ISBN 978-85-4301-381-7.

CAUSES of Database Failure: What to do in a Data Recovery Emergency. [*S. l.*], c2021. Disponível em: https://platinumdatarecovery.com/blog/causes-of-database-failure-what-to-do-in-a-data-recovery-emergency. Acesso em: 15 maio 2021.

CRASH recovery. [*S. l.*], c 2021. Disponível em: https://www.ibm.com/docs/en/db2/11.5?topic=recover-crash-recovery. Acesso em: 15 maio 2021.

GUIRRA, Michelle. **Controle de concorrência entre transações em bancos de dados**. [*S. l.*], 2013. Disponível em: https://www.devmedia.com.br/controle-de-concorrencia-entre-transacoes-em-bancos-de-dados/27756. Acesso em: 16 maio 2021.

DEBATIN, Lucas. **Segurança e Recuperação**. [S.L]: Univali, 2021. 50 slides, color.