|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | http://www.devmedia.com.br/imagens/casper.gif | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | |  |  |  | | --- | |  |   **Recuperando um banco de dados SQL Server após um desastre**   |  | | --- | |  |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  |  | | --- | --- | | [http://www.devmedia.com.br/imagens/fotoscolunistas/PspDevMedia.bmp](http://www.devmedia.com.br/space.asp?id=158574) | [**PAULO SéRGIO PEREIRA** Paulo Sergio Pereira (psergio.p@terra.com.br) Bacharel em Ciência da Computação pela Univap (Universidade do Vale do Paraíba) é desenvolvedor Java, Progress, Visual Basic, Visual C++.NET e ADVPL. É administrador de bancos de dados Progress, SQL Server 2000/2005 e DB2. Atua como desenvolvedor pela empresa compasso (www.compasso.com.br) em projetos Visual Basic, Java e Oracle.](http://www.devmedia.com.br/space.asp?id=158574) | | | |  | | **Recuperando um banco de dados SQL Server após um desastre**      Em qualquer profissão é sempre importante tomar cuidado com as ferramentas de trabalho, no ramo da tecnologia de informação o cuidado deve ser em dobo. Imagine que um banco de dados é a vida da empresa, perdendo o banco de dados a situação fica complicada. Neste artigo mostrarei como utilizar o transaction log do SQL Server para situações onde o banco de dados é corrompido. Montaremos passo a passo uma situação em que o banco de dados é corrompido e só temos em mãos um backup full antigo e o transaction log.    **Entendendo o transaction log**    O transaction log é utilizado para gravar as alterações efetuadas antes que estas venham a ser gravadas definitivamente no arquivo de dados. Todos os comandos submetidos ao SQL Server são gravados no transaction log (na verdade o que fica armazenado no transaction log são apenas os comandos submetidos ao SQL Server), mesmo que o usuário omita os comandos BEGIN TRANSACTION  e END TRANSACTION o qual informa ao SQL Server que os comandos contidos no bloco devem ser tratados como uma transação. O transaction log não deve ser colocado no mesmo disco onde ficará o arquivo de dados pois fazendo isso além de ter uma queda na performance você não terá como recuperar seu banco de dados após a ocorrência de um desastre pois caso o disco venha a apresentar problemas ambos os arquivos poderão ser perdidos.  Após receber um comando que altera informações o SQL carrega as páginas que contem as informações para a memória, em seguida o comando é gravado no transaction log e a página alterada em memória, de tempos em tempos é executado um processo de checkpoint o qual salva as páginas alteradas para o disco e marca a transação no transaction log como concluída.  Um banco de dados pode ter três diferentes maneiras de armazenamento no transaction log que são: Full, Simple e Bulk-Logged, veremos as 2 primeiras.  1.    Full : Nesta opção todas as operações são armazenadas no transaction log sendo removidas somente quando um backup full é feito ou quando o usuário executar comandos para que o log seja truncado;  2.    Simple: Esta opção não pode ser usada em nosso exemplo pois as transações gravadas no log são apagadas sempre que ocorre um checkpoint, sendo assim o usuário depende sempre do backup full para recuperar as informações;    **Criando um banco de dados para exemplo**    Vamos agora criar m banco de dados para utilizar em nosso exemplo, na **Figura 1** você pode instrução completada para a criação do banco. Perceba que o arquivo de dados e o transaction log foram colocados em discos diferentes pois caso o arquivo de dados venha a corromper-se por falha no disco o transacion log estará integro em outra unidade.    21-05-2007.JPG  **Figura 1.** Criando banco de dados para exemplo    **Alterando o recovery model do nosso banco de dados**    Após a criação vamos agora alterar a opção recovery model para full em nosso banco de dados para que as informações contidas no transaction log não sejam removidas após o checkpoint. Veja na **Figura 2** a tela para alteração da opção. Em seguida vamos fazer um backup full da base mesmo não tendo nenhuma informação ainda. Veja na **Figura 3** a instrução para backup do banco de dados.    21-05-2007pic02.JPG  **Figura 2.** Alterando a opção recovery model para full    **21-05-2007pic03.JPG**  **Figura 3.** Fazendo um backup FULL do banco de dados    **Criando e populando a tabela de exemplo**  Vamos agora criar um tabela no banco de dados e inserir alguns registros, em seguida simularemos uma falha de disco que irá corromper o nosso arquivo de dados. Veja na **Figura 4**os comandos para a criação da tabela e na **Figura 5** a inclusão de alguns registros.    21-05-2007pic04.JPG  **Figura 4.** Criação de tabela de exemplo    **21-05-2007pic05.JPG**  **Figura 5.** Populando a tabela de exemplo    **Simulando a falha de disco e perda do arquivo de dados**    Vamos agora simular uma falha de disco que ira corromper o arquivo de dados, os procedimentos que seguiremos são os seguintes:  1.    Parar o serviço do SQL Sever;  2.    Mover o arquivo de dados (.mdf) para outro local e colocar em seu lugar um arquivo texto zerado com a extensão .mdf;  3.    Carregar novamente o SQL Server;  Após os procedimentos acima você deve acessar o Enterprise manager, você irá verificar que o banco de dados Teste está marcado como “*Suspect*”, isto significa que o SQL não pôde verificar a integridade do banco de dados e provavelmente algum dano ocorreu e será necessário restaurar a base de dados. Veja a **Figura 6**.    **21-05-2007pic06.JPG**  **Figura 6.** O banco de dados foi marcado como “*Suspect*”    **Recuperando o banco de dados**    Chegamos agora ao ponto final, agora veremos como recuperar as informações contidas no log de transações e que foram incluídas após o ultimo backup full da base de dados. Devemos seguir os seguintes passos:  1.    Fazer um backup do log de transações mesmo o banco estando como suspect (lembrando que o log de transações deve estar integro em outra unidade de disco);  2.    Em alguns casos é necessário a exclusão do banco marcado como “*Suspect*”;  3.    Restaurar o ultimo backup full;  4.    Restaurar o backup do log recém criado;  Pronto, apesar da falha de disco você consegue recuperar as informações segundos antes da falha. Recomendo que você trate-se o transaction log com cuidado, ele pode salvar sua vida.    **Conclusão**    O transaction log é um recurso extremamente importante, se você coloca-lo em uma unidade que não a mesma do arquivo de dados a probabilidade de perda de informações tende a zero. É claro que isto não é uma pratica que deve sempre ser usada, pois uma estratégia de backup deve ser adotada sempre. Neste artigo procurei mostrar uma situação onde a falha inesperada pode causar a perda de informações de horas ou até dias (após o ultimo backup) porém usando o transaction log você pode recuperar informações em um horário muito próximo da falha.  [***www.devmedia.com.br/articles/viewcomp.asp?comp=5452***](www.devmedia.com.br/articles/viewcomp.asp?comp=5452) | |  | |  | |  | |  | | http://www.devmedia.com.br/imagens/casper.gif | |  | |  |  |