SISTEMA GERENCIADOR DE BANCO DE DADOS

AULA 2

Prof. Leonel da Rocha

CONVERSA INICIAL

Nesta abordagem, veremos as funcionalidades de administração de um sistema gerenciador de banco de dados. Veremos como alterar e atualizar a estrutura de um database, tarefa realizada pelo administrador de banco de dados, conhecido como *DBA*, para agilizar a utilização do banco, permitindo também o seu crescimento e facilidades na sua administração.

Trataremos de itens como as tabelas do sistema, que são estruturas criadas automaticamente para a administração de um banco de dados que devem ser conhecidas e dominadas pelo administrador para possíveis implementações e alterações.

Discutiremos a realização e administração das cópias de segurança, que chamamos de *backup*, suas características e as preocupações que o administrador deve ter com essas cópias. Mostraremos como as cópias devem ser planejadas e os tipos que podem ser realizados, que são os *backups* full, ou seja completo, e os parciais. Na esteira das cópias de segurança, falaremos sobre recuperação dos dados, comumente chamado de *restore*, que é o retorno dos dados devido a algum problema apresentado, como tabelas ou discos corrompidos, ou ainda a criação de um novo banco de dados com conteúdo salvo por um *backup*.

TEMA 1 – BANCO DE DADOS

Um banco de dados é uma coleção de dados organizados que se relacionam entre si de maneira a criarem algum sentido, transformando-se em informação, proporcionando maior eficiência para uma pesquisa ou estudo. Se mostram muito importantes para as organizações, transformando-se em uma das principais peças dos sistemas de informação.

Os Sistemas Gerenciadores de Banco de Dados, ou SGBD, possuem um conjunto de requisitos funcionais que são implementados em um banco de dados, que são: segurança, integridade, controle de concorrência, e recuperação e tolerância a falhas.

A seguir, veremos detalhadamente as funcionalidades de um SGBD:

Segurança: refere-se às medidas de proteção utilizadas para proteger os dados contra acessos não autorizados, para preservar a confidencialidade, integridade e disponibilidade dos dados. Para a uma efetiva aplicação dessas

medidas, existem boas práticas de segurança de dados que incluem técnicas de proteção de dados, como: criptografia, gerenciamento de controles, suporte a edição, subconjunto e mascaramento de dados, auditoria e monitoramento.

Os dados são um dos ativos mais importantes para as organizações. Por isso devem ser protegidos de qualquer tipo de acesso não autorizado. Quando há violações de dados, auditorias sem sucesso e falhas no cumprimento de requisitos regulatórios, existe a possibilidade de resultar em danos à imagem da organização, perda de valor da marca, comprometimento da propriedade intelectual, multas por não conformidade, além de perdas financeiras significativas, muitas vezes comprometedoras, inclusive da continuidade das atividades da organização. Os dados confidenciais incluem informações de identificação pessoal, financeiras, de saúde e propriedade intelectual. Diante desses aspectos, os dados devem ser protegidos evitando sua violação objetivando o alcance da conformidade.

O controle de acesso aos dados é uma maneira indispensável para implementar a proteção aos dados, a validação da identidade do usuário que acessará o banco de dados, sua autenticação e o controle de quais operações ele poderá executar, sua autorização. Quando bem implementado, esses controles de autenticação e autorização auxiliam na proteção dos dados diante de invasores. Outro fator importante sobre controle de acesso é implementar a divisão de tarefas para os usuários do banco, pois isso evita que usuários com acesso de administrador façam uso dessa condição para acessar dados confidenciais, podendo evitar, além disso, alterações acidentais ou mesmo com más intenções ao banco de dados.

Para incrementar a segurança é importante implementar formas de auditoria e monitoramento para todas as atividades do banco de dados. Incluindo as atividades na rede onde os servidores de banco estão usando, além das atividades de login realizadas diretamente no servidor, realizada por administradores, pois essas atividades geralmente ignoram qualquer monitoramento de rede. Lembrando que a auditoria deve registrar as ações mesmo quando a rede for criptografada. A auditoria para esses casos deve implementar controles fortes e abrangentes, incluindo informações sobre os dados, os usuários, local de realização da solicitação, data e hora da operação e o comando SQL que foi enviado ao banco.

Outro fator importante de segurança é a proteção dos bancos de dados na nuvem. Se as implementações em nuvem reduzem custos, agilizam os

processos, diminuem a sobrecarga de trabalho da TI, trazem ao mesmo tempo riscos adicionais, como uma abrangência de rede maior, administradores desconhecidos e infraestrutura compartilhada. Nesses casos se forem empregadas as boas práticas de segurança de banco de dados, o ambiente em nuvem pode fornecer melhor segurança do que as redes privadas, reduzindo custos como já vimos e aumentando a agilidade de todo o processo de informações.

Integridade: refere-se ao fato dos dados de um banco devem estar de acordo com o que eles representam no mundo real. Em uma organização onde todos os dados estão armazenados em um SGBD e sem autorização ou razão são alterados ou excluídos, com certeza provocará muitos problemas. É para evitar os impactos que esse tipo de ação pode causar aos negócios que a integridade de dados é tratada como assunto primordial. A integridade é referente ao quanto os dados são confiáveis e consistentes ao longo do seu ciclo de vida útil. Tem como prioridade preservar o conhecimento para que nada seja comprometido ou perdido. A integridade dos dados é o foco principal de muitas políticas de segurança, por isso diversas ferramentas de proteção são implementadas para que os dados sejam preservados e íntegros ao extremo.

Para a implementação da integridade de dados existem dois tipos: física e lógica. A integridade lógica ainda por ser subdividida em mais quatro categorias. Vamos ver a seguir esses tipos de implementação de integridade de dados:

- Integridade física: é a proteção total dos dados, garantindo sua precisão, à medida em que são armazenados e recuperados. Ela pode ser comprometida em casos de desastres ou quando algum invasor danifica o banco ou o deixa fora de serviço. Outro fator como erros humanos e falhas de acesso também impossibilitam que os usuários tenham dados precisos.
- Integridade lógica: é o tipo de integridade implementada pelo SGBD que garante que os dados não sejam alterados de forma incorreta. Isso pode prevenir falhas contra erros operacionais e de possíveis ataques, obviamente quando as são devidamente implementadas.

A integridade lógica é subdividida em quatro categorias que veremos a seguir:

- Integridade da entidade: em um banco de dados relacional, existem colunas, linhas que por sua vez formam as tabelas. Para as informações

serem precisas, muitas vezes o número de registros a serem consultados é grande, porém a extração dessas informações não deve trazer nada além do que foi solicitado. E não deve ocultar informações.

- Integridade referencial: controles que garantem que os dados armazenamento em uma tabela são válidos baseados em dados armazenados em outra tabela. Com as regras incorporadas ao banco de dados, bem como à criação de chaves de acesso, assegura-se que sejam feitas somente as alterações, adições e exclusões corretas.

As regras podem incluir restrições que: não permitem a inclusão de dados duplicados; forçam que os dados incluídos são precisos; não permitem a inclusão de dados incorretos.

- Integridade de domínio: são controles que asseguram a precisão de cada parte dos dados de um domínio. Limitam os valores que serão aceitos em cada coluna de uma tabela. Determina se a coluna aceitará somente números, data ou caracteres. Regras que limitam o formato e a quantidade de informações são inseridas nesse tipo de integridade. Se um campo, por exemplo, controlar datas, valores diferentes desse formato não serão aceitos.
- Integridade definida pelo usuário: quando as integridades de entidade, referencial e domínio não são suficientes para proteger os dados, o usuário poderá criar suas próprias regras para atender as necessidades particulares e garantir que os dados sejam protegidos e as regras atendidas.
- É importante manter a integridade de dados pois quando for necessário tomar uma decisão, será preciso se basear nos dados disponíveis. Mas quando eles não são precisos e apresentam duplicidades, é provável que a decisão seja equivocada, podendo provocar prejuízos, seja para a organização como um todo ou para o responsável pelo processo.

Como vimos, a integridade dos dados impacta diretamente na tomada de decisão, pois quanto mais claros, confiáveis e consistentes os dados são, maiores são as chances de uma decisão ser acertada. Ela é importante para garantir o acesso aos dados no local correto e no tempo certo, pois quando o acesso não é preciso, de pouco a informação vai valer, comprometendo todo o processo decisório.

A integridade de dados pode ser comprometida de várias maneiras, existindo diversos fatores que podem afetar a integridade dos dados armazenados, dentre eles os mais comuns são:

- Erro dos usuários: manipulação de dados de forma incorreta, onde as regras de negócio não são respeitadas;
- Erros de replicação: os dados não são transferidos de um banco para o outro;
- Vírus e ataques: vírus e hackers estão a todo momento tentando invadir um computador para alterar, excluir ou roubar dados;
- Problemas com *Hardware*: crashes e problemas de funcionamento de dos dispositivos envolvidos com o banco de dados comprometem a integridade dos dados de diversas maneiras, como: dificuldades de acesso, lentidão e problemas de segurança.

Controle de concorrência: quando temos um banco de dados que é utilizado por mais de um usuário, é necessário administrar o controle de concorrência entre os dados que serão acessadas pelos usuários. Esse controle de concorrência é necessário quando dois ou mais usuários tentam acessar o mesmo repositório de dados, daí é realizado um controle entre essas transações para gerenciar essa concorrência. Existindo para isso diversas técnicas de controle de concorrência que são usadas para assegurar a propriedade de não interferência entre as operações, ou o isolamento das transações executadas simultaneamente. Essas técnicas utilizam a serialização, que é a execução das transações de forma serial. É importante entender que transações são todas as operações executadas entre o início e o fim da transação, e que para gerenciar as transações é preciso conhecer as propriedades ACID (acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade) que são utilizadas pelos métodos de controle de concorrência e recuperação do SGBD:

A seguir veremos as propriedades ACID, que são importantes nesse processo de controle de concorrência:

- A Atomicidade de uma transação é o princípio que ela é uma unidade de processamento atômica, isso quer dizer que a transação deve ser realizada por completo do contrário ela não deve ser realizada. Caso aconteça alguma falha durante o processo, as alterações parciais desta transação devem ser desfeitas, garantindo assim a integridade dos dados.
- A Consistência dos dados é garantida pelo fato de uma transação ser

executada do início até o fim sem interferência de outras transações, ou seja, é a execução de uma transação isolada, levando o banco de um estado consistente para outro.

- O Isolamento é o fato de uma transação parecer isolada de outras, mesmo com várias transações executadas simultaneamente. O SGBD garantirá que apenas uma transação seja executada por vez e que a próxima só comece quando a atual for finalizada.

- A Durabilidade ou permanência dos dados é a garantia de que as mudanças que ocorreram ao término de uma transação que foi finalizada com sucesso persistam no banco. Isso quer dizer que após uma operação ser encerrada com êxito, os dados gravados devem permanecer no banco de dados mesmo que ocorram falhas no sistema.



Crédito: thinkhubstudio/Shutterstock.

TEMA 2 – TABELAS DO SISTEMA

Tabelas base do sistema são tabelas auxiliares que armazenam dados de controle de um banco de dados. O banco de dados *master* é contém tabelas que são encontradas apenas em sua estrutura. Essas tabelas contêm os registros de todos os objetos dos outros bancos de dados criados em um SGBD esses dados de controle são conhecidos como *metadados*.

Um usuário que possua permissão de *CONTROL, ALTER* ou *VIEW DEFINITION* em um banco de dados pode ver os metadados da tabela base do sistema na exibição do conteúdo da tabela sys.objects. O usuário poderá consultar os nomes e as identificações dos objetos das tabelas do banco de dados usando o comando para selecionar o conteúdo das colunas OBJECT_NAME e OBJECT_ID.

Em um SGBD é possível fazer referências às colunas nas tabelas do

sistema. O único problema é que muitas das colunas nas tabelas do sistema não são visíveis para os usuários. Portanto, não devem ser escritos aplicativos que consultem diretamente essas colunas. Para recuperar informações armazenadas nas tabelas do sistema, os aplicativos devem usar os seguintes componentes: Procedimentos armazenados do sistema; Instruções e funções Transact-SQL; SQL Server Management Objects (SMO); RMO (Replication Management Objects); Funções de catálogo da API do banco de dados.



Crédito: PopTika/Shutterstock.

TEMA 3 – GERENCIAMENTO DINÂMICO

Views de gerenciamento dinâmico (DMV) são um conjunto de informações que mostram para o DBA o comportamento do ambiente do banco de dados, como informações sobre índices, espaço ocupado pelos objetos, consultas dos usuários que consumem mais recursos e outras situações que auxiliam na administração do SGBD.

Esse tipo de informação é importante para o DBA que têm por objetivo administrar todo o ambiente, se possível antecipando problemas que podem acontecer no SGBD. Com informações contidas nas DMV, o DBA poderá elaborar um log dos problemas ocorridos dentro do banco de dados, podendo salvar essas informações, até mesmo, em um banco de dados gerencial.

Gerenciar o ambiente de um SGBD, mantendo o banco de dados com o melhor desempenho possível e atuando de forma proativa são algumas das responsabilidades de um DBA. Para essas responsabilidades serem realizadas com sucesso e da melhor maneira possível, cada SGBD traz um conjunto de

objetos de sistema que mostram aos seus administradores informações importantes sobre o comportamento do sistema operacional e até mesmo informações sobre o desempenho das *queries* mais lentas que estão rodando no banco de dados, possibilitando que toda a equipe da TI possa trabalhar para resolver todos os problemas encontrados. Em relação as *queries*, pode-se alterar o seu código ou criando índices para resolver problemas de desempenho.

Com muitas informações à disposição dos DBAs, é possível criar scripts para agrupar informações semelhantes e controlar dessa maneira o ambiente por meio do conhecimento gerado por essas informações que muitas vezes se repetem e podem ser resolvidos de uma maneira semelhante a outras situações ocorridas. É possível fazer integrações com outras ferramentas, criar *dashboards* com base na situação atual de vários ambientes, realizando atualizações periódicas das informações capturadas.

TEMA 4 – *BACKUP* COMPLETO E PARCIAL

Vamos ver qual é a importância e os benefícios de se fazer *backup* de um banco de dados, em que situações eles devem ser feitos e a sua restauração. Definir estratégias de *backup* e restauração além da preocupação com a segurança das mídias onde eles são realizados.

Os componentes de *backup* e *restore* de um SGBD oferecem uma proteção essencial para os dados armazenados. Minimizando dessa maneira o risco de perda de dados em virtude de algum tipo de problema, é necessário fazer *backup* dos dados para preservar as modificações feitas e isso deve ser feito regularmente. Planejar estratégias de *backup* e *restore* é importante para proteger os bancos de dados contra perda de dados e é importante ter em mente que uma estratégia de restauração dos *backups* será primordial para que quando qualquer tipo de falha acontecer seja possível recuperar o banco de dados eficientemente. Nesse ponto podemos levantar uma questão relativa a realização dos *backups*: o *restore* funcionará quando for preciso fazer a restauração de um banco de dados??? Os SGBD nos mostram que realizaram o *backup* com sucesso, por isso é possível fazer checklists dessas atividades diariamente, mas o ponto de interrogação é sobre a restauração, ou *restore,* dos dados de um banco. Por quê? Porque de nada adiantará o *backup* ser realizado corretamente conforme a estratégia estabelecida mas por alguma razão técnica ele não puder ser utilizado independente da razão que isso venha a acontecer.

Então precisamos ter em mente que é importantíssimo termos *backup* dos dados, mas é ainda mais importante temos certeza de que o *restore* desses dados acontecerá corretamente. Para que isso aconteça sem problemas, além da estratégia de *backup*, devemos implementar também uma estratégia de *restore* dos dados, realizando simulações de *restore* periodicamente. Elas não precisam acontecer na mesma frequência dos *backups* mas devem ser feitas regularmente e de uma forma organizada e padronizada. Podemos ter em mente que o *backup* é importante, mas restaurar os dados é mais importante ainda!!!

Vamos ver agora a importância da realização de um *backup* dos bancos de dados de um SGBD, a execução de procedimentos de restauração de teste desses *backups* e o armazenamento das mídias de *backups* em um local seguro evitará a perda dos dados, que pode gerar uma situação catastrófica. Realizar *backup* é a única maneira de proteger os dados, mas também devemos ter certeza de que é possível restaurar esses dados.

Os *backups* válidos de um banco de dados, possibilitam a recuperação dos dados de diversos problemas, tais como:

- Falhas de *hardware*, seja um disco danificado ou perda de um servidor;
- Falhas humanas, que modificam objetos e dados por engano;
- Ataques cibernéticos;
- Desastres naturais.

Os *backups* também podem ser úteis para fins administrativos, como copiar um banco de dados de um servidor para outro, configurar o espelhamento do banco de dados, além de fazer arquivamento para determinados períodos, como fechamento mensal ou anual, folha de pagamento e outros procedimentos em que seja necessário ter um congelamento dos dados.

Para realizarmos o *backup* e a restauração dos dados é importante termos uma estratégia bem definida para essas duas ações. O *backup* e a restauração devem ser planejados em um ambiente específico e devem funcionar com recursos disponíveis. Quando essa estratégia é bem projetada, ela equilibra os requisitos de negócios para máxima disponibilidade de dados e mínima perda de dados, considerando também o custo de manutenção e armazenamento de *backups*.

Uma estratégia bem planejada de *backup* e *restore* deve contemplar as duas partes. A estratégia de *backup* definirá o responsável pela execução, o tipo e a frequência, tipo de mídia, *hardware* exigido para execução, como os testes

serão realizados, local de armazenamento da mídia de *backup*, como ela deve ser armazenada e qual a segurança necessária. Por sua vez, a estratégia de restauração definirá o responsável, como elas devem ser executadas para atender às metas de disponibilidade e minimizar as perdas dos dados, como as restaurações serão testadas, frequência das restaurações e *hardware* de retorno para os testes.

O desenvolvimento de uma estratégia de *backup* e restauração exige planejamento, implementação e testes minuciosos. E o que precisamos entender que os testes são muito importantes e necessários, pois não existe uma estratégia de *backup* eficaz até que não se tenha testado a restauração com sucesso, levando em consideração todas as possibilidades incluídas na estratégia de restauração e também a consistência física e o acesso do banco de dados restaurado. Para o desenvolvimento dessa estratégia é necessário considerar uma variedade de fatores, tais como:

- As metas para os bancos de dados de produção, relacionados aos requisitos de disponibilidade e proteção contra perda dos dados;
- A natureza operacional de cada banco de dados: tamanho, padrões de utilização, fonte do conteúdo e os requisitos dos dados;
- Limitações de recursos: *hardware*, humano, local de armazenamento da mídia de *backup* e sua segurança física.

Para o desenvolvimento de uma estratégia de *backup* e *restore* eficiente algumas recomendações devem ser seguidas:

Armazenar a mídia de *backup* do banco de dados em um local físico ou dispositivo separado dos arquivos originais. Quando o *hardware* que armazena o banco de dados falha, a recuperação depende do acesso à mídia de *backup*.

Para planejar uma estratégia de *backup* alguns fatores devem ser considerados:

- Disponibilidade diária do banco de dados;
- Se existir um período de pouca atividade previsível, agendar o *backup* completo para esse horário;
- Frequência das alterações e atualizações.
- Para alterações frequentes, considerar os seguintes pontos:
- No modelo de recuperação simples, agendar *backups* diferenciais entre os *backups* completos. Salientando que um *backup* diferencial captura só as alterações feitas desde o último *backup* completo;

- No modelo de recuperação completa, agendar *backups* de log frequentes. O agendamento de *backups* diferenciais entre *backups* completos pode reduzir o tempo de restauração reduzindo o número de *backups* de log a serem restaurados após a restauração dos dados.
- Conforme os *backups* se tornam antigos, o risco de perda de dados é maior. Antes de descartar os *backups* antigos, é preciso considerar se há necessidade de recuperação de um tempo muito anterior dos dados;
- Para implementar uma estratégia de *backup* e restauração é preciso calcular quanto espaço em disco um *backup* de banco de dados completo irá usar. A operação de *backup* copia os dados do banco para o arquivo de *backup*, mas apenas os dados reais, portanto, o *backup* é geralmente será menor do que o banco de dados.

Dando continuidade ao plano estratégico de *backups*, um item importante para esse planejamento é o agendamento dos *backups*. Lembrando que a execução do *backup* tem um efeito pequeno sobre as transações do banco de dados, portanto, o *backup* pode ser feito durante a utilização do banco pelos usuários. Depois da definição dos tipos de *backups* e a frequência de execução de cada tipo, é recomendável agendar os *backups* como parte de um plano de manutenção para o banco de dados.

A estratégia de *backup* não estará completa se a restauração não for testada. Então é preciso ter em mento que é importante testar a estratégia completa de *backup* para cada um dos bancos de dados, restaurando uma cópia em um sistema de teste. É necessário testar a restauração de cada tipo de *backup* realizado: full, diferencial e de log. Também é recomendável que, depois de restaurar o *backup*, sejam executadas verificações de consistência do banco de dados para validar se a mídia de *backup* não com problemas.

Por último, mas não menos importante, é preciso documentar a estratégia de *backup* e restauração que será adotada. É interessante documentar os procedimentos de *backup* e restauração, mantendo uma cópia dessa documentação em um relatório de tarefas. É recomendável manter um manual de operações para cada banco de dados. O manual operacional deve documentar o local e horário dos *backups*, o nome dos dispositivos de *backup* e o tempo necessário e o local para restauração dos *backups* de teste.



Crédito: Alexander Supertramp/Shutterstock.

TEMA 5 – RECUPERAÇÃO DE UM BANCO DE DADOS

Para um sistema de banco de dados, a rotina de *backup* é essencial para a proteção dos dados. Mas como nós já vimos anteriormente, o mais importante e, geralmente deixado de lado, é a restauração desses dados. Porque o objetivo principal do *backup* é a proteção dos dados e se necessário recuperá-los após algum problema para poder acessá-los novamente.

Diante desse fator importantíssimo, a restauração dos dados é uma parte fundamental das estratégias de *backup*. Para não ter problemas quanto a este procedimento, vamos estudar alguns pontos importantes para não ter problemas quando for necessário retornar o conteúdo de um banco de dados. Vamos lá!

A rotina de copiar os dados é chamada de *backup* e é composta por duas etapas fundamentais. A primeira é quando os dados são copiados para serem armazenados em um espaço seguro e preferencialmente diferente da sua origem. Já a segunda acontece quando é necessário recuperar esses dados por eventuais problemas acontecidos na fonte ou por motivos administrativos, tais como replicação de dados, criação de base de desenvolvimento, entre outros. Podemos dizer que a primeira copia e protege, a segunda recupera e transfere.

A restauração de dados é a ação de recuperar os dados armazenados em um dispositivo durante o procedimento de *backup*, garantindo que os dados estejam salvos e disponíveis para eventual utilização. Muitas estratégias de *backup* se preocupam apenas em criar uma cópia de segurança, sem testar a restauração desses dados no caso de desastres. E isso pode ser tão preocupante quanto não ter cópia nenhuma dos dados,

Como podemos supor, isso pode ser um grande problema, pois a sensação de segurança e tranquilidade pode se transformar em um pesadelo se quando for necessário retornar os dados, esse retorno não funcionar. Isso pode causar impactos no negócio da organização trazendo sérios riscos, inclusive para a continuidade dos negócios. Podemos dizer que pior do que não tem *backup* é ter e não poder restaurar os dados.

A relação entre a rotinas de *backup* e *restore* é muito próxima. A única maneira de garantir que os dados possam restaurados é por meio da realização das cópias de segurança. Porém, essa relação de cópia e restauração não é o único cenário que devemos nos preocupar. Vamos imaginar uma situação hipotética, em um site de vendas, que tem uma estratégia de realização dos *backups* implementa e funcionando perfeita. Porém, sem nunca ter feito nenhum teste de retorno. Em determinado momento acontece algum tipo de falha no servidor e o site fica indisponível. É providenciado outro servidor imediatamente, pois é necessário disponibilizar o site novamente. Nesse momento é necessário recriar o banco de dados com todos os dados armazenados até o momento da falha do servidor. Quando o responsável questiona a área de TI sobre o *backup*, obtém um retorno dizendo que a cópia do banco levará mais de um dia para ser restaurada ou, pior ainda, que a cópia de segurança está corrompida e não poderá ser restaurada. Em ambos os casos o prejuízo é certo, seja pela demora ou pela incapacidade de recuperar os dados. O problema é que a extensão do prejuízo pode ser crucial para a continuidade dos negócios.

Para evitar situação como essa, a restauração dos dados é fundamental em qualquer estratégia de TI e não se pode criar uma estratégia de *backup* no qual a restauração não seja eficiente, já que os dados não poderão ser utilizados de forma apropriada em um momento de necessidade.

Temos que ter algo em mente quando precisamos criar uma rotina de *backup* e restauração de dados, é que a eficiência dessa rotina não pode ser alcançada por meio de produtos, ferramentas ou de uma tecnologia. Contudo, para que ela seja eficiente e possa ser cumprir seus objetivos, é preciso focar no processo, monitoramento constantemente das rotinas de *backup*, gerenciamento das cópias de segurança, coleta informações para análise e procedimentos para testar todo esse processo, desde a cópia até o retorno dos dados.

Será necessário desenvolver uma expertise em toda a rotina de *backup* e os pontos que a formam, de modo a garantir que o cliente final possa ter acesso às suas cópias de segurança com a qualidade que necessita no momento oportuno

e sem falhas. Os responsáveis pelas rotinas de *backup* devem entender que o processo de *backup* e o *restore* de dados vai muito além do que simplesmente fazer apenas uma cópia dos dados. Será necessário gerenciar todo o processo, com verificações e validações diversas, além de testes de restauração agendados e aprovados por todos que fazem parte do processo: gerencia, técnicos e usuários.

Uma das maneiras de garantir que o *backup* e o *restore* sejam realizados de uma maneira correta e eficaz é o monitoramento. Será necessário não apenas manter um gerenciamento constante sobre as rotinas de *backup*, mas também fazer testes rotineiramente para garantir a eficiência do sistema. Portanto, apenas com um olhar atento e próximo é que se conseguirá atingir a excelência no gerenciamento de dados.

Podemos analisar o cenário do site de vendas, onde se tivesse sido realizado algum teste de restauração dos dados, teria sido possível verificar que o tempo para a restauração era muito longo. Com essa informação em mãos, seria possível elaborar uma estratégia para contornar essa vulnerabilidade, sem deixar que o problema fosse descoberto apenas quando foi necessário recuperar os dados em uma situação real.

Para criar uma estratégia de *backup* alguns pontos devem ser levados em consideração para as definições das rotinas. Nem todas as organizações utilizarão os mesmos parâmetros e modelos, pois as demandas são diferentes. Veremos a seguir alguns pontos relevantes na estruturação de uma estratégia de *backup*. Vamos lá!

Primeiro ponto a se considerar é o volume de dados pois o tempo necessário para a cópia e a restauração é proporcional à quantidade de dados em questão. Também é uma operação limitada à capacidade operacional e técnica dos dispositivos e sistemas utilizados, em que se deve considerar a velocidade de deslocamento dos discos de armazenamento, velocidade de *download* e upload, velocidade da rede e outras situações relacionadas ao ambiente em que a estratégia será aplicada. Por isso é imprescindível conhecer essas especificações para determinar o tempo total da operação, tanto para copiar como para restaurar.

Outro ponto a ser considerado na elaboração da estratégia de segurança é a relevância dos dados, pois em um sistema, nas suas diversas operações, toda interação gera dados, porém nem todas as alterações têm relevância para a continuidade das operações. Portanto, é importante fragmentar os dados que são

mais importantes para o sistema, dessa forma poderemos dar prioridade para os dados com maior relevância aos negócios e estabelecer uma estratégia diferente do restante dos dados. Isso permitirá, talvez, uma agilidade maior tanto para copiar quanto para restaurar os dados.

Estabelecendo esse tipo de prioridade, pode ser possível garantir uma seletividade maior dos objetos que são constantemente atualizados e estabelecer uma estratégia de *backups* diferenciadas, possibilitando a realização de *backups* completos do banco em intervalos maiores e as alterações em tempo menor, permitindo dessa maneira, que os dados cruciais sejam constantemente copiados, atualizando de forma segura as informações.

Sobre segurança dos dados, existe um conceito sobre a estruturação dos *backups* de onde é recomendado a disponibilização do *backup* por meio da regra 3-2-1: onde se estabelece que devemos ter 3 cópias de segurança, distribuídas em pelo menos 2 tipos de armazenamento diferentes; sendo 1 delas fora do ambiente onde estão os dados originais.

A lógica por trás dessa regra, onde é recomendado rotinas de precaução básica, com planos distintos para cada uma das situações. Primeiro as três cópias de segurança. Segundo, o armazenamento em dois formatos diferentes, que podem variar entre mídias físicas (HDs e SSDs) ou virtuais, como os serviços de cloud. Terceiro, são consideradas as catástrofes naturais ou acidentes possam inviabilizar o acesso ao armazenamento físico quando recomenda que uma das cópias de segura fique fora do ambiente onde estão os dados originais. Como podemos verificar, a regra 3-2-1 é simples, mas se mostra eficiente por propor soluções que garantam o acesso a cópia dos dados mesmo que problemas mais sérios aconteçam.



Crédito: Gorodenkoff/Shutterstock.

FINALIZANDO

Nesta abordagem, tratamos as funcionalidades de administração de um sistema gerenciador de banco de dados, com alteração e atualização da estrutura de um banco de dados, tarefa realizada pelo DBA, para agilizar a sua utilização, permitindo também o seu crescimento e facilidades na sua administração.

Vimos as tabelas do sistema, que são estruturas criadas automaticamente para a administração de um banco de dados que devem ser conhecidas e dominadas pelo DBA para possíveis implementações e alterações. Por último, trabalhamos um item importantíssimo que é a realização e administração dos *backups*, suas características e as preocupações que o administrador deve ter com essas cópias de segurança.

Vimos como as cópias devem ser planejadas e os tipos que podem ser realizados, completo e parcial. Na esteira das cópias de segurança, tratamos sobre recuperação dos dados, ou o *restore*, que é o retorno dos dados devido a algum problema apresentado, como tabelas ou discos corrompidos, ou ainda a criação de um novo banco de dados com conteúdo salvo por um *backup*.

REFERÊNCIAS

BARRIE, H. **Dominando Firebird: Uma Referência para Desenvolvedores de Bancos de Dados**. ED. Ciência Moderna, 2006.

CARLOS, S., **Comparativo de desempenho de Bancos de dados de Código Aberto**. (UFPE), 2011.

CARNEIRO, A. **Técnicas de Otimização de Bancos de Dados** - Um estudo Comparativo: MySQL e PostegreSQL. (FURG), 2011.

LEITE, M. **Acessando Bancos de Dados com Ferramentas RAD**. Braspor, Ed. 2007.

SILBERSCHATZ, A., KORT; SUDARSHAM. **Sistemas de Bancos de Dados**. Quinta ED. 2006.