Estratégia Concursos

Aula 00

SGBDs para Concursos (Área de TI) - Curso Regular

Autor:

Equipe Informática e TI, Thiago Rodrigues Cavalcanti

03 de Fevereiro de 2023

Índice

1) Administração de Dados e Banco de Dados 3

Teoria de SGDBs **2**
Administradores de banco de dados x Administradores de dados **2**
Visão Geral das responsabilidades **3**
Design do banco de dados *3*
Seleção e Avaliação dos Sistemas de Gerenciamento de Banco de Dados *4*
Monitoramento de performance e tuning *5*
Garantido a disponibilidade *7*
Segurança e autorização *7*
Backup e recuperação *9*
Garantia das regras de governança *10*
Garantia da integridade dos dados *11*
Gerenciamento de Gatilhos e Procedimentos Armazenados *13*
Considerações Finais **15**

THIAGO CAVALCANTI
PROFESSOR

TEORIA DE SGDBS

Nessa aula vamos fazer uma introdução aos conceitos e ações associadas ao contexto de administração de dados e banco de dados. Nessa altura do campeonato você já deveria conhecer essas duas figuras que fazem parte da gestão de uma grande base de dados. Se você, por acaso, se esqueceu ... vamos aproveitar e fazer uma rápida revisão desse tópico. Veremos ainda que a administração de banco de dados deve:

- Controlar bancos de dados usando instruções SQL em prol da segurança e integridade.
- Reconhecer a importância de gerenciar gatilhos e procedimentos armazenados.
- Compreender o processo para escolher e avaliar SGBDs.
- Possuir uma visão sobre os ambientes de processamento nos quais a tecnologia de banco de dados é utilizada.

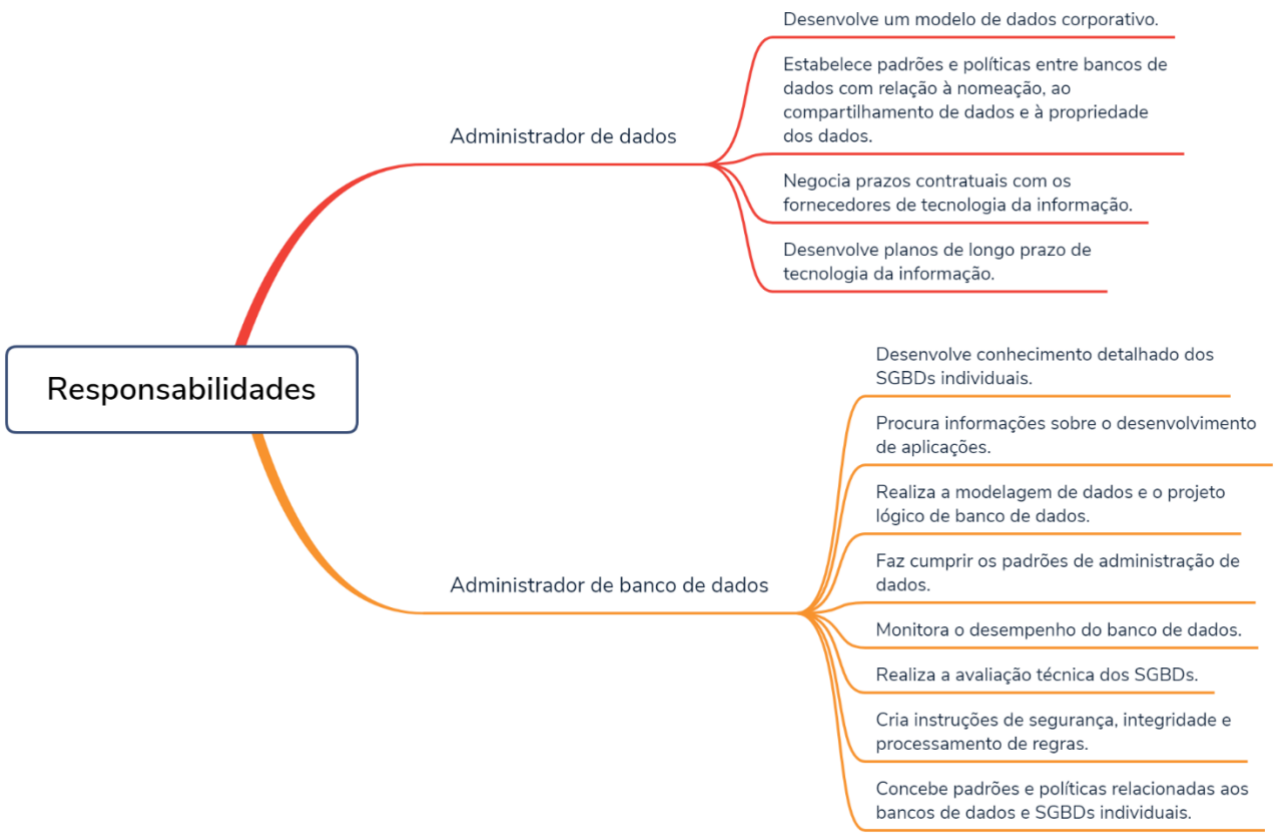
ADMINISTRADORES DE BANCO DE DADOS X ADMINISTRADORES DE DADOS

São duas as principais responsabilidades gerenciais que existem dentro do controle de recursos de informação. O administrador de dados (DA - *data administrator*) é uma **posição da média ou alta gerência** que tem amplas responsabilidades pelo gerenciamento de recursos de informação. O administrador de banco de dados (DBA - *database administrador*) desempenha uma **função de suporte** cujas responsabilidades estão relacionadas aos bancos de dados individuais e aos SGBDs.

A figura abaixo compara as responsabilidades de ambas as funções. O administrador de dados vê os recursos de informação em um contexto mais amplo que o administrador de banco de dados. O primeiro considera todos os tipos de dados, estejam eles armazenados em bancos de dados relacionais, arquivos, páginas Web ou fontes externas. O segundo normalmente considera apenas os dados armazenados nos bancos de dados.

Um **modelo de dados corporativo** é um **modelo conceitual** de dados de uma organização. O desenvolvimento de um modelo de dados corporativo é uma das responsabilidades mais importantes do administrador de dados. Ele oferece um modelo integrado de todos os bancos de dados da organização.

Para cumprir as responsabilidades mencionadas, os administradores de banco de dados usam **uma variedade de ferramentas**. São ferramentas usadas para **modelagem de dados**, projeto lógico de banco de dados, projeto físico de banco de dados, criação de gatilhos e procedimentos armazenados. Algumas das ferramentas são estruturam instruções SQL (CREATE VIEW e CREATE INDEX), ao passo que outras fazem parte das ferramentas CASE para o desenvolvimento de banco de dados.



O principal objetivo desta aula é apresentar as responsabilidades do DBA. A próxima sessão tenta apresentar noções gerais sobre essas responsabilidades sem focar em um SGBD específico.

VISÃO GERAL DAS RESPONSABILIDADES

O DBA deve ser capaz de realizar muitas tarefas para **garantir que os dados da organização e o banco de dados sejam úteis, utilizáveis, disponíveis e corretos**. Essas tarefas incluem design, monitoramento de performance e tuning, garantia da disponibilidade, autorizações de segurança, backup e recuperação, garantia da integridade dos dados, ou, qualquer outra ação que faça parte da interface com as bases de dados da empresa. Vamos examinar cada um destes tópicos.

DESIGN DO BANCO DE DADOS

A primeira tarefa que a maioria das pessoas imagina de quando pensam o trabalho de um DBA é a capacidade de criar bancos de dados. Esses bancos devem ser especialmente bem desenhados. Para projetar e criar corretamente bancos de dados relacionais, os DBAs devem entender e fazer uso das boas práticas de design relacional.

Eles devem entender a **teoria relacional** e a **implementação específica** do SGBD que está sendo utilizado para criação do banco de dados. O design de banco de dados requer uma sólida

compreensão das técnicas de **modelagem de dados conceitual e lógica**. Sendo assim, a capacidade de criar e interpretar diagramas entidade-relacionamento (ER) é essencial para a concepção de um banco de dados relacional.

Além disso, o DBA deve ser capaz de **transformar um modelo de dados lógicos em um modelo de dados físicos**. O DBA deve assegurar que a concepção e implementação de banco de dados permitirão a criação de um banco de dados útil para as aplicações e os clientes. Em resumo a primeira tarefa importante do DBA:

O DBA deve ser capaz de transformar um modelo de dados numa base de dados lógica física.

Na verdade, o projeto de um banco de dados é uma habilidade importante para o DBA. No entanto, o trabalho do DBA é desproporcionalmente associado com o projeto de banco de dados. Apesar da atividade de projetar bancos de dados ser importante, é uma parte relativamente pequena do trabalho do DBA. O DBA provavelmente vai gastar mais tempo **administrando e ajustando os bancos de dados** do que originalmente na concepção e construção.

De maneira nenhuma, porém, você deve interpretar isto dizendo que o design de banco de dados não é importante. Um projeto relacional pobre pode resultar num desempenho fraco, ou em um banco de dados que não atende as necessidades da organização e com dados potencialmente imprecisos.

Seleção e Avaliação dos Sistemas de Gerenciamento de Banco de Dados

A seleção e avaliação de um SGBD pode ser uma tarefa extremamente importante para uma organização. Os SGBDs formam uma parte essencial da infraestrutura de computação. Visto que as organizações estão empenhadas em conduzir comércio eletrônico pela Internet e a extrair valor dos bancos de dados operacionais, os SGBDs desempenham um papel cada vez mais importante.

O processo de seleção e avaliação envolve uma estimativa detalhada das necessidades da organização e dos recursos de um SGBD que está sendo pleiteado. A meta do processo é determinar um pequeno conjunto de sistemas candidatos que serão investigados mais detalhadamente. Em virtude de seu nível de detalhamento, o DBA executa a maior parte das tarefas. Portanto, ele precisa conhecer meticulosamente o SGBD para dar cabo de tudo isso.

Após a conclusão do processo de seleção e avaliação, os dois ou três principais SGBDs candidatos devem ser avaliados mais detalhadamente. Podem ser usados **benchmarks** (padrões de desempenho/marca de referência) para obter uma avaliação mais pormenorizada dos SGBDs candidatos. **Um benchmark é uma carga de trabalho para avaliar o desempenho de um sistema ou produto**.

Um bom benchmark deve ser **relevante, portável, escalável e compreensível**. Visto que para desenvolver bons benchmarks a competência é essencial, a maioria das organizações não deve tentar desenvolver seu próprio benchmark. Felizmente, o Conselho de Processamento de Transações (***Transaction Processing Council – TPC***) já desenvolveu inúmeros benchmarks padrão, de domínio específico, de acordo com o resumo apresentado na tabela a seguir.

Benchmark	Descrição	Medidas de Desempenho
TPC-C	*Benchmark* de entrada de pedido *on-line*	Transações por minuto; preço por transação por minuto
TPC-H	Suporte a decisões para consultas *ad hoc*	Consultas compostas por hora; preço por consulta composta por hora
TPC-App	Transações *business-to-business* com aplicações e serviços Web	Interações de serviços Web por segundo (*service interation per second* – SIPS), por servidor de aplicação; SIPS total; preço por SIPS
TPC-W	*Benchmark* de comércio eletrônico	Interações Web por segundo; preço por interação Web por segundo

Resumo dos TPCs

Cada benchmark foi desenvolvido no decorrer de um longo período, com base nas informações obtidas de um grupo de vários colaboradores. O DBA pode usar os resultados do TPC para cada benchmark a fim de obter estimativas razoáveis sobre o desempenho de um SGBD particular, em um ambiente de hardware/software específico. Os resultados de desempenho do TPC envolvem o desempenho total do sistema, não apenas o do SGBD.

A fase final do processo de seleção pode envolver considerações não técnicas por parte do DA, bem como da gerência sênior e da equipe jurídica. A estimativa das perspectivas futuras de cada fornecedor é essencial porque os sistemas de informação podem ter uma vida longa. Por exemplo, se o SGBD escolhido não progredir no compasso do setor, talvez não suporte iniciativas futuras e atualizações nos sistemas de informação que o utilizam. Em decorrência dos altos custos fixos e variáveis (taxas de manutenção) de um SGBD, a negociação geralmente é um fator fundamental do processo de seleção final.

MONITORAMENTO DE PERFORMANCE E TUNING

O segundo papel importante associado com o DBA é o **monitoramento e ajuste** de desempenho, também conhecido como *tuning*. Mas o que queremos dizer com desempenho do banco de dados? Pense, por um momento, em desempenho do banco de dados usando os conceitos familiares de oferta e demanda.

Usuários exigem informações do banco de dados. O SGBD fornece as informações para aqueles que as solicitam. A taxa de entrega (informação por alguma medida de tempo) à qual o SGBD fornece as informações pode ser denominada de desempenho do banco de dados. Lógico que não é tão simples assim, mas essa é a ideia inicial que você precisa ter em mente.

Cinco fatores influenciam o desempenho do banco de dados: **carga de trabalho (*workload*)**, **rendimento (*throughput*)**, **recursos**, **otimização** e **contenção**.

A **carga de trabalho** que é solicitado do SGBD define a demanda. É uma combinação de transações on-line, batch jobs, consultas ad hoc, consultas analíticas e comandos feitos diretamente ao sistema a qualquer momento. A carga de trabalho pode variar drasticamente ao longo do tempo.

Às vezes a carga de trabalho pode ser prevista (como processamento pesado no final do mês de folha de pagamento, ou um acesso mais leve após as 20h, quando a maioria dos usuários já não está nas estações de trabalho), mas em outros momentos, é imprevisível. A carga de trabalho global tem um grande impacto no desempenho do banco de dados.

A taxa de transferência ou ***throughtput*** define a capacidade total de processamento do hardware e software. É um composto pela velocidade de I/O, velocidade da CPU, capacidades paralelas da máquina, e pela eficiência do sistema operacional e do próprio SGBD. As ferramentas de hardware e software à disposição do sistema são conhecidas como os **recursos do sistema**. Exemplos incluem o *kernel* do banco de dados, espaços em disco, controladores de cache, e o microcódigo.

O quarto elemento definidor do desempenho do banco de dados é a **otimização**. Todos os tipos de sistemas podem ser otimizados, mas as consultas relacionais são as únicas tarefas em que a otimização é realizada internamente ao SGBD. No entanto, existem muitos outros fatores que precisam ser otimizados (definições SQL, parâmetros de banco de dados, programação mais eficiente, e assim por diante) para permitir que o otimizador crie os caminhos de acesso mais eficientes.

Quando a demanda (carga de trabalho) para um determinado recurso é alta, a **contenção** pode acontecer. Contenção é a condição em que dois ou mais componentes da carga de trabalho estão tentando usar um único recurso de uma forma conflitante (por exemplo, duas atualizações para o mesmo dado). Quando a contenção aumenta, o rendimento diminui.

---

*Por conseguinte, o desempenho do banco de dados pode ser definido como a otimização da utilização dos **recursos** para aumentar o **rendimento** e minimizar a **contenção**, que permita a maior **carga de trabalho** possível ser processada.*

Sempre que os problemas de desempenho são encontrados por um aplicativo que usa um banco de dados, o DBA é geralmente o primeiro a ser chamado para resolver o problema. Claro que, o DBA não pode controlar o desempenho do banco de dados no vácuo. Aplicações se comunicam regularmente com outras aplicações, sistemas e componentes da infraestrutura de TI.

Um efetivo monitoramento de desempenho e estratégia de ajuste não requer apenas uma perícia sobre o SGBD, mas o conhecimento fora do âmbito da administração de banco de dados. Muitas tarefas de gerenciamento de desempenho devem ser compartilhadas entre o DBA e outros técnicos. Em outras palavras, lidar com problemas de desempenho é realmente um esforço de toda a empresa.

O DBA deve estar vigilante no monitoramento do desempenho do sistema, do banco de dados e das aplicações. Tanto quanto possível, isto deve ser conseguido utilizando software e scripts

automatizados. *Polling* de tabelas do sistema e construção de alertas baseados em limites podem ser usados para identificar problemas de forma proativa.

Os alertas podem ser configurados para disparar um e-mail ao DBA quando as métricas de desempenho não estiverem dentro dos limites aceitos. Muitas tarefas e habilidades são necessárias ao DBA para assegurar o acesso às bases de dados eficientes.

Algumas destas habilidades incluem a construção de índices apropriados, especificação de buffers e caches, alinhando da implementação de banco de dados com a infraestrutura de TI, monitoramento contínuo das bases de dados e aplicações, recuperação de banco de dados e adaptação às mudanças de negócios. Alguns fatores que podem afetar são mais dados, mais usuários, processamento adicional, e mudanças de requisitos ou regulamentação.

GARANTIDO A DISPONIBILIDADE

Disponibilidade de dados e bancos de dados é muitas vezes estreitamente alinhada com o desempenho, mas na verdade é uma preocupação separada. Claro, se o SGBD está *off-line*, o desempenho vai ser horrível, porque os dados não podem ser acessados. Mas, garantir a disponibilidade do banco de dados é um processo multifacetado.

O primeiro componente de disponibilidade é manter o SGBDs instalado e funcionando. Monitoramento e alertas automáticos podem ser usados para alertar sobre falhas do SGBD e executar alguma ação corretiva.

Bancos de dados individuais também devem ser mantidos de modo que os dados neles contidos estejam disponíveis sempre que as aplicações e os clientes exigirem. Isso requer do DBA um projeto do banco de dados para que ele possa ser mantido com interrupções mínimas, mas também para ajudar aplicativos para minimizar os conflitos quando o acesso simultâneo é necessário.

Um componente adicional de disponibilidade é minimizar a quantidade de tempo de inatividade necessário para executar tarefas administrativas. Quanto mais rápido o DBA executar tarefas administrativas que exigem que os bancos de dados sejam desligados, mais disponíveis os dados se tornam.

Cada vez mais, vendedores de SGBD e fornecedores de software independentes (ISVs) estão fornecendo utilitários que podem ser executadas em bancos de dados enquanto as aplicações leem e escrevem, sem interrupções. Mas estes geralmente requerem mais habilidade e planejamento por parte do DBA.

O DBA deve compreender todos esses **aspectos de disponibilidade** e garantir que cada aplicativo esteja recebendo o nível correto de qualidade de serviço, ou seja, adequado as suas necessidades.

SEGURANÇA E AUTORIZAÇÃO

Uma vez que o banco de dados foi concebido e implementado, programadores e usuários precisarão acessar e modificar os dados no banco de dados. Mas só os programadores e usuários autorizados

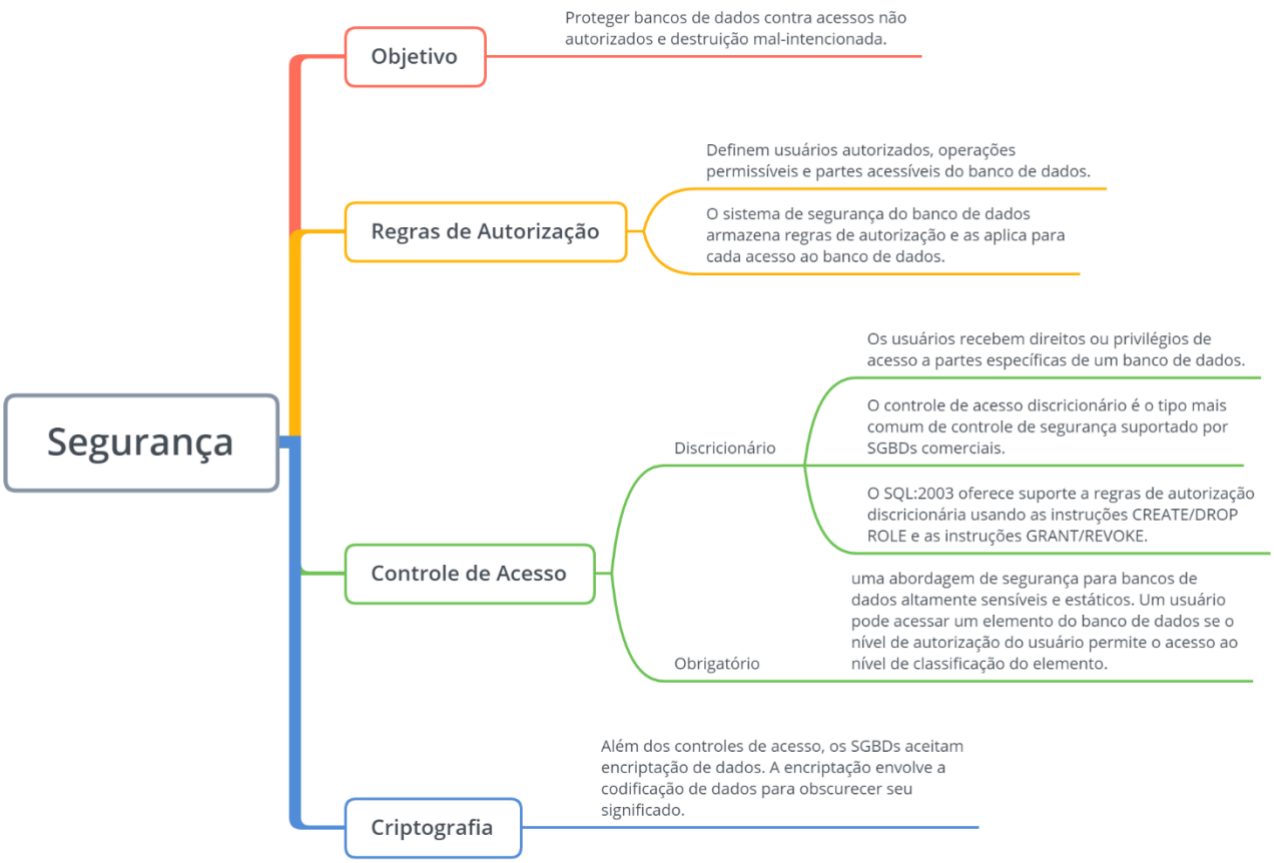
devem ter acesso para evitar violações de segurança e modificação de dados inadequada. É da responsabilidade do DBA garantir que os dados estão disponíveis apenas para usuários autorizados.

Normalmente, embora nem sempre, o DBA trabalha com as características internas de segurança dos SGBD sob a forma dos comandos de SQL GRANT e REVOKE, bem como quaisquer características do DBMS para autorização grupo. A segurança deve ser administrada por muitas ações exigidas pelo ambiente de banco de dados:

- Criação de objetos de banco de dados, incluindo bancos de dados, tabelas, *views* e estruturas de programa
- A alteração da estrutura de objetos de banco
- Acessar o catálogo do sistema
- Leitura e modificação de dados em tabelas
- Criar e acessar funções definidas pelo usuário e tipos de dados
- Executar procedimentos armazenados
- Iniciar e param os bancos de dados
- Definir parâmetros e especificações do ambiente SGBD
- Execução de utilitários de BD como LOAD, RECOVER e REORG

A **segurança de banco de dados** pode ser aplicada de outras maneiras também. Por exemplo, podem ser criadas visões para bloquear colunas ou linhas sensíveis, não permitindo que os dados sejam visualizados por usuários finais e programadores. O DBA também interage frequentemente com métodos de segurança externos quando eles afetam a segurança do banco de dados (ex.: LDAP).

O DBA deve compreender e ser capaz de implementar qualquer aspecto de segurança que afeta o acesso aos bancos de dados. Uma área que deve ser de interesse particular, levando em consideração as violações de dados, é ataques de injeção SQL e como evitá-los. Enfim, o DBA deve compreender todos os aspectos da segurança que afetam o acesso a bancos de dados. Abaixo apresentamos um resumo dos conceitos de segurança em banco de dados.



BACKUP E RECUPERAÇÃO

O DBA deve estar preparado para **recuperar os dados** em caso de falhas. Uma falha pode significar diferentes coisas, desde um problema de sistema ou erro de programa a um desastre natural que aconteça em uma determinada organização. A maioria das recuperações ocorre como um resultado de erro do software de aplicação ou de erro humano.

As falhas de hardware não são tão relevantes como costumavam ser. De fato, as estimativas dos analistas indicam que 80 por cento dos erros de aplicação são devido a falhas de software ou erro humano. O DBA deve estar preparado para recuperar o banco de dados a um estado consistente, não importa qual a causa, e a fazer tal operação o mais rapidamente possível.

O primeiro tipo de recuperação de dados que normalmente vem à mente é uma **recuperação ao estado atual,** geralmente acontece quando uma falha tira o sistema do ar. O resultado da recuperação é que o banco de dados é trazido de volta ao seu estado atual no momento da falha. As aplicações ficam completamente indisponíveis até que a recuperação esteja completa.

Outro tipo de recuperação tradicional é uma **recuperação point-in-time**. A recuperação deste tipo normalmente é realizada para lidar com um problema em nível de aplicação. As técnicas convencionais para executar uma recuperação point-in-time irão remover os efeitos de todas as

transações desde um ponto específico no tempo. Isso pode trazer problemas se aconteceu alguma transação válida durante esse período que ainda precisam ser aplicadas.

Recuperação de transações é um terceiro tipo de recuperação que aborda as deficiências dos tipos tradicionais de recuperação: o **tempo de inatividade** e **perda de dados de boa qualidade**. Assim, a recuperação de transação é uma recuperação de aplicativos em que os efeitos das transações específicas durante um período especificado são removidos do banco de dados.

Portanto, a recuperação de transação é por vezes referida como a recuperação de aplicações. Perceba que esse tipo de recuperação serve para ajustar a base de dados depois de modificações inconsistentes feitas por aplicações.

A maioria dos DBAs pensa em recuperação como meio apenas para resolver os desastres, como falhas de hardware. Apesar das falhas de hardware ainda ocorrem, e dos gestores de banco de dados precisarem estar preparados para se recuperar de tais falhas, a maioria das recuperações são necessárias, como já falamos, devido a erro humano ou erros nos programas.

O DBA deve estar preparado para lidar com todos esses tipos de recuperação. Isto envolve o desenvolvimento de uma estratégia de **cópia de segurança** para assegurar que os dados não sejam perdidos no caso de um erro em software, hardware, ou um processo manual.

A estratégia deve ser aplicável ao processamento de banco de dados, por isso deve incluir cópias de imagens de arquivos de banco de dados, bem como um plano de backup/recuperação para os logs de banco de dados. Ele precisa dar conta de qualquer atividade de arquivos que possam impactar as aplicações de banco de dados também.

GARANTIA DAS REGRAS DE GOVERNANÇA

Assegurar o cumprimento das **regulamentações da indústria e governamentais** é uma tarefa adicional da administração de banco de dados, pelo menos em termos de implementação de controles adequados (auditoria). O DBA deve trabalhar com gestores, auditores e especialistas de negócio para compreender a normatização que se aplica a sua área e a maneira pela qual os dados devem ser tratados.

Certos aspectos da conformidade com os padrões regulatórios afetam diretamente o trabalho do DBA. Por exemplo, os **regulamentos podem conter procedimentos de segurança e autorização específicos**, requisitos de auditoria, especificações de backup de dados e alteração nos procedimentos de gestão. Para garantir o cumprimento, no entanto, uma documentação mais rigorosa pode ser exigida, ou talvez maior diligência ou de automação (tais como rastreamento de auditoria mais detalhado).

Outros aspectos de conformidade regulatória podem exigir que o DBA adote diferentes técnicas, táticas e habilidades. Por exemplo, os regulamentos de retenção de dados podem necessitar que as informações fossem mantidas por muito tempo além do necessário numa base de dados de produção, o que requer capacidades de arquivamento do banco de dados. Ou determinados dados que podem necessitar de proteção adicional, necessitando de mascaramento de dados ou, em alguns casos, de criptografia.

Os DBAs não devem ser incumbidos de entender os regulamentos em qualquer profundidade, nem devem estabelecer padrões pelos quais a organização está em conformidade com os regulamentos. No entanto, os DBAs irão se envolver para ajudar a definir os controles e procedimentos para projetos de conformidade, especificamente e especialmente no que diz respeito ao tratamento de dados.

GARANTIA DA INTEGRIDADE DOS DADOS

Um banco de dados deve ser projetado para armazenar os dados corretos no caminho correto sem que os dados se tornem danificados ou corrompidos. Para assegurar esse processo, o DBA implementa as regras de integridade usando recursos do SGBD. Três aspectos da integridade são relevantes para nossa discussão de bases de dados: **físico, semântico** e **interno**.

Problemas físicos podem ser manipulados usando recursos do SGBD como domínios e tipos de dados. O DBA escolhe o tipo de dados apropriado para cada coluna de cada tabela. Esta ação assegura que apenas os dados desse tipo são armazenados no banco de dados. Isto é, o SGBD impõe a integridade dos dados em relação ao seu tipo.

Uma coluna definida como "integer" pode conter apenas números inteiros. As tentativas de armazenar valores não numéricos ou não inteiros em uma coluna definida como inteiro falhará. O DBA também pode usar restrições para delinear com maior profundidade do tipo de dados que pode ser armazenado no banco de dados para cada coluna. A maioria dos produtos de SGBDs relacionais fornecem os seguintes tipos de restrições:

- **Restrições referenciais** são usadas para especificar as colunas que definem quaisquer relações entre tabelas. Restrições referenciais são usados para implementar a integridade referencial, o que garante que todas as referências de dados em uma coluna (ou conjunto de colunas) de uma tabela são válidos em relação aos dados em outra coluna da mesma ou de uma tabela diferente.
- As **restrições de unicidade** garantem que o valor para uma coluna ou um conjunto de colunas ocorra apenas uma vez na tabela.
- As restrições ***check*** são usadas para colocar as regras de integridade mais complexas em uma coluna ou conjunto de colunas de uma tabela. As restrições de verificação são tipicamente definidas utilizando o SQL e podem ser utilizadas para definir os valores de dados que são permitidas para uma coluna ou um conjunto de colunas.

Um tipo de dado indica a espécie de dado (caractere, numérico, sim/não etc.) e operações permissíveis (operações numéricas, operações com string etc.) para colunas que estão usando o tipo de dado. No SQL:2003, a capacidade para definir novos tipos de dados por meio da instrução CREATE DOMAIN é limitada. É possível criar um domínio **como um subconjunto de um tipo de dado padrão**. O exemplo a seguir mostra a instrução CREATE DOMAIN, bem como o uso de novos domínios em lugar de tipos de dados padrão. A cláusula CHECK define uma restrição para que o domínio limite os seus valores a um subconjunto do tipo de dados padrão.

Instruções CREATE DOMAIN e o Uso de Domínios (SQL:2003)

```
CREATE DOMAIN TurmaAluno AS CHAR(2)
CHECK (VALUE IN ('FR', 'SO', 'JR', 'SR'));

CREATE DOMAIN CargaHorariaCurso AS SMALLINT
CHECK (VALUE BETWEEN 1 AND 9);
```

Integridade semântica é mais difícil de controlar e menos facilmente definida. Os DBAs devem estar preparados para aplicar políticas e práticas para garantir que os dados armazenados em seus bancos de dados sejam precisos, apropriados e utilizáveis. Um exemplo de uma questão semântica é a qualidade dos dados na base de dados.

Armazenar todos os dados que reúne as definições de integridade físicas especificadas para o banco de dados não é suficiente. Procedimentos e práticas precisam estar no local para garantir a qualidade dos dados. Por exemplo, uma base de dados de clientes que contém um endereço errado ou número de telefone para 25 por cento dos clientes nela armazenada é um exemplo de uma base de dados com fraca qualidade.

Não existe um método sistemático ou físico de assegurar a precisão dos dados. A qualidade dos dados é incentivada por meio de código adequado da aplicação, práticas empresariais sólidas e políticas de dados específicas. Redundância é outra questão semântica. Se elementos de dados são armazenados de forma redundante em todo o banco de dados, o DBA deve documentar esse fato e trabalhar para garantir que os procedimentos estão no lugar para manter os dados redundantes sincronizadas e precisas.

O aspecto final da integridade é uma questão interna do SGBD. O SGBDS depende de estruturas internas e código para manter as ligações, ponteiros e identificadores. Na maioria dos casos o SGBD vai fazer um bom trabalho de manter estas estruturas, mas o DBA precisa estar ciente da sua existência e como lidar quando o SGBD falhar. Integridade interna é essencial nas seguintes áreas:

- **Consistência de índices**. Um índice é realmente nada além de uma lista ordenada de ponteiros para dados em tabelas de banco de dados. Se por algum motivo o índice fica fora de sincronia com os dados, o acesso indexado pode falhar ao retornar os dados incorretos. O DBA tem ferramentas à sua disposição para procurar e corrigir esses tipos de erros.
- **Consistência de ponteiros**. Às vezes, grandes objetos multimídia não são armazenados nos mesmos arquivos físicos como outros dados. Portanto, o SGBD requer estruturas de ponteiro para manter os dados de multimídia sincronizados com os dados da tabela base. Mais uma vez, esses ponteiros podem ficar fora de sincronia se os procedimentos de administração adequados não forem seguidos.
- **Consistência de backup**. Alguns produtos de SGBD, ocasionalmente, levam cópias de segurança impróprias que efetivamente não podem ser usados para a recuperação. É essencial para identificar esses cenários e tomar ações corretivas.

No geral, garantir a integridade é uma habilidade essencial DBA.

GERENCIAMENTO DE GATILHOS E PROCEDIMENTOS ARMAZENADOS

Embora o DBA escreva os gatilhos e procedimentos armazenados para ajudar a gerenciar os bancos de dados, sua principal responsabilidade é gerenciar gatilhos e procedimentos armazenados, e não os escrever. É também sua responsabilidade estabelecer padrões para as práticas de codificação, monitorar dependências e compreender as interações entre os gatilhos.

Em relação às práticas de codificação, o DBA deve considerar **os padrões de documentação**, **o uso de parâmetros** e **o conteúdo**. Os padrões de documentação devem incluir padrões, explicações sobre os parâmetros e descrição das condições prévias e posteriores dos procedimentos. O uso de parâmetros em procedimentos e funções deve ser monitorado. As funções devem usar apenas parâmetros de entrada e não ter efeitos secundários.

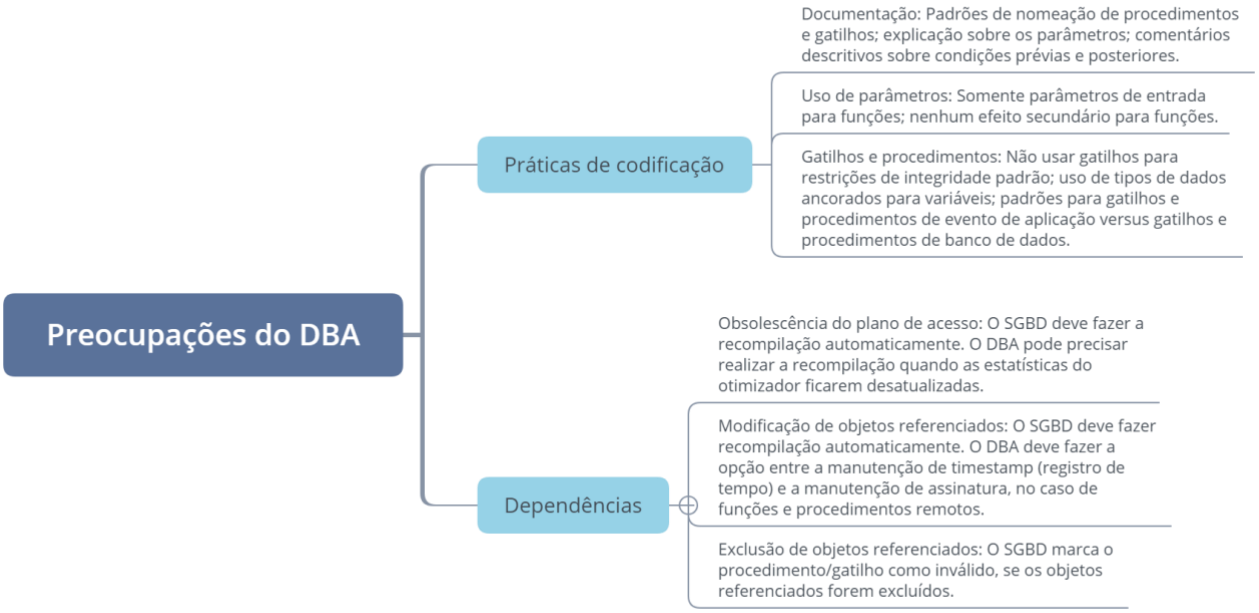
Com respeito ao conteúdo, os gatilhos não devem realizar a verificação de integridade que possa ser codificada como restrições de integridade declarativa (restrições CHECK, chaves primárias, chaves estrangeiras etc.). Para diminuir a manutenção, os gatilhos e procedimentos armazenados devem fazer referência aos tipos de dados das colunas associadas do banco de dados.

O gatilho ou procedimento armazenado depende das tabelas, das visões, dos procedimentos e dos papéis que ele menciona, bem como de **planos de acesso criados pelo compilador do SQL**. Quando um objeto mencionado muda, seus dependentes devem ser recompilados. A maioria dos SGBDs mantém dependências para assegurar que os gatilhos e procedimentos armazenados funcionem corretamente. Se um procedimento ou gatilho usar uma instrução SQL, a maioria dos SGBDs automaticamente recompilará o procedimento ou gatilho, se o plano de acesso correspondente ficar desatualizado.

O DBA deve estar atento para as limitações do SGBD e das ferramentas fornecidas para o gerenciamento de dependência. Em relação aos planos de acesso, o DBA deve compreender que a **recompilação manual pode ser necessária** se as estatísticas do otimizador ficarem desatualizadas. No caso de procedimentos e funções armazenados remotamente, o DBA pode escolher entre a manutenção de dependência *timestamp* e de assinatura.

Na manutenção de *timestamp*, o SGBD recompilará um objeto dependente para qualquer alteração nos objetos referenciados. Essa manutenção pode provocar um excesso de recompilação porque muitas alterações nos objetos referenciados não exigem recompilação dos objetos dependentes. A manutenção de assinatura requer recompilação quando uma assinatura (nome ou uso do parâmetro) muda.

O DBA deve também estar atento para o fato de que o SGBD não recompilará um procedimento ou gatilho se um objeto referenciado for excluído. O procedimento ou gatilho dependente será marcado como inválido porque a recompilação não é possível. Vejamos as preocupações relevantes do DBA organizadas didaticamente no mapa mental abaixo:


Preocupações do DBA
Práticas de codificação
Documentação: Padrões de nomeação de procedimentos e gatilhos; explicação sobre os parâmetros; comentários descritivos sobre condições prévias e posteriores.
Uso de parâmetros: Somente parâmetros de entrada para funções; nenhum efeito secundário para funções.
Gatilhos e procedimentos: Não usar gatilhos para restrições de integridade padrão; uso de tipos de dados ancorados para variáveis; padrões para gatilhos e procedimentos de evento de aplicação versus gatilhos e procedimentos de banco de dados.
Dependências
Obsolescência do plano de acesso: O SGBD deve fazer a recompilação automaticamente. O DBA pode precisar realizar a recompilação quando as estatísticas do otimizador ficarem desatualizadas.
Modificação de objetos referenciados: O SGBD deve fazer recompilação automaticamente. O DBA deve fazer a opção entre a manutenção de timestamp (registro de tempo) e a manutenção de assinatura, no caso de funções e procedimentos remotos.
Exclusão de objetos referenciados: O SGBD marca o procedimento/gatilho como inválido, se os objetos referenciados forem excluídos.

CONSIDERAÇÕES FINAIS

Prezados alunos

Chegamos, pois, ao final da sobre administração de banco de dados!

Espero reencontrar você cada vez mais motivado com o assunto.

Espero que tenha gostado! E até breve!

Thiago Cavalcanti

ESSA LEI TODO MUNDO CONHECE:
PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.

1. Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.

2. Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).

3. Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).

4. Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).

5. Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).

6. Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).

7. Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.

8. O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.

Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.