



Administrando Banco de Dados



fundaçāo bradesco | escola virtual

Sumário

Apresentação	5
Módulo 1	
Administrador de banco de dados	6
Administrador de banco de dados	7
O profissional e o mercado	7
Tarefas de um processo de administração de banco de dados	10
Carreiras em banco de dados próximas do DBA	15
Importância dos dados na gestão	16
Bancos de dados	17
SQL	24
O SQL e seus componentes	26
Ferramentas de gestão do SQL	27
Módulo 2	30
Arquitetura de um SGBD (Sistema Geral de Banco de Dados)	30
Arquitetura de um SGBD (Sistema Geral de Banco de Dados)	31
Modelos de dados	31
Esquemas (<i>schemas</i>)	32
Instâncias	35
Explorando um banco de dados relacional	39
Tabelas do sistema	41
Tipos de arquivos	43
Criar um banco de dados	45

Sumário

Módulo 3	49
Gerenciamento e manutenção de sistemas de dados	49
Gerenciamento e manutenção de sistemas de dados	50
Manutenção de banco de dados	50
Anexar, desanexar, importação, exportação e backup de dados- um exercício	53
Anexar um banco de dados	53
Desanexar um banco de dados	54
Criando BD e tabelas	55
Importar dados via SSMS	56
Exportar dados	56
Backup	58
Backup diferencial	60
A ferramenta de segurança Docker	61

Sumário

Módulo 4	64
Procedimentos administrativos	64
Procedimentos administrativos	65
Logons e usuários	65
Arquivos de <i>log</i>	67
Monitoramento de desempenho e de atividade de servidor	68
Gestão da segurança	70
Gestão de segurança no nível do banco de dados	71
Chaves, certificados e criptografia	71
Auditoria de segurança	74
Fechamento	76
Referências	77

Apresentação

Olá!

Bem-vindo(a) ao curso **Administrando Banco de Dados**.

Este curso traz conhecimentos que poderão inseri-lo na carreira de administrador de banco de dados. Aqui você entenderá as diferenças entre os diversos profissionais que atuam em atividades ligadas à gerência de bancos de dados e quais são os tipos mais utilizados e adequados a empresas de qualquer porte.

Veremos uma introdução sobre como administrar um banco de dados e, desta forma, conseguir realizar um bom gerenciamento das atividades que estão ligadas a essa administração. Além disso, você conhecerá os conceitos que fazem parte do dia a dia de um administrador de banco de dados, podendo colocar em prática esse conjunto indexado de informações.

Apresentaremos também os Modelos de dados, SQL (*Structured Query Language*), esquemas, instâncias, tabelas e todo esse fascinante universo que envolve os bancos de dados.

Vamos começar esta jornada?



Módulo 1

Administrador de banco de dados

Administrador de banco de dados

A partir de agora, você terá acesso às atividades que permeiam o cotidiano de um administrador de banco de dados, as competências necessárias, os modelos, a arquitetura e o seu papel no dia a dia das organizações.

Desejamos que seus estudos sejam proveitosos e que o impulsionem em seu crescimento profissional!

O profissional e o mercado

Você já parou para refletir que todo o conhecimento estudado pode ser concretizado na realidade de diversas formas, ou seja, que ele pode ser aplicado no seu dia a dia? Dentre essas inúmeras maneiras, é possível usá-lo no chamado **mercado de trabalho**, sendo oferecido, principalmente, às empresas privadas.

Assim, preparamos o vídeo, a seguir, que detalha essa relação entre profissional e mercado. Acompanhe!



Vídeo

Confira o [vídeo](#) sobre sobre o profissional e o mercado dentro da área de administração de banco de dados.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Quando nos referimos ao banco de dados, falamos de conjuntos de dados ou informações relevantes para um modelo de negócio. Isso significa que, teoricamente, o aglomerado de dados em uma organização seria um “banco de dados”. No entanto, no cotidiano das empresas, as próprias dinâmicas comerciais e de serviços exigem que os dados e as informações sejam dispostos de forma estruturada e normativa.

Para isso, essa estrutura deve se organizar em um Sistema Gerenciador de Banco de Dados, ou SGBD.

O SGBD é um software capaz de promover um gerenciamento aos acessos realizados em um banco de dados específico. Isso constitui um processo ou agrupamento de processos no interior de uma estrutura, criada para a tarefa de armazenamento e acesso às informações disponíveis em um banco.

Para isso, alguém precisa ser o responsável por esta organização, sendo este o DBA (*Data Base Administrator*), ou seja, o administrador de banco de dados, ou ainda, gerenciador de banco de dados. Trata-se do profissional que utiliza os programas computacionais para armazenar e organizar os dados de uma empresa, que mais tarde serão consultados.

Esses dados podem conter informações diversas, como:

- informações financeiras;
- pedidos de clientes;
- movimentação de insumos e equipamentos; entre outras.

E os administradores de banco de dados são os profissionais que vão garantir que esses dados estejam disponíveis para as pessoas autorizadas de uma organização consultarem.

Sendo de responsabilidade desse profissional, também, compreender com quais tipos de dados uma organização necessita trabalhar. Inclusive em determinando a maneira como esses dados serão gerenciados e escolhendo o gerenciador de banco de dados adequado para o desempenho do sistema.

Por isso, é recomendável que o administrador tenha especialização em algum SGBD, visto que ele precisa conhecer uma base mínima das características, utilização e gerenciamento de SGBDs diferentes. Ele também deverá criar estratégias de monitoria nas atividades diárias de um sistema e, no caso de problemas identificados, prover a solução de forma ágil na sua correção. Com isso, os DBAs têm a responsabilidade de implementar medidas e políticas de segurança, pois são os responsáveis por impedir acessos não autorizados.

Dessa forma, as informações de um banco de dados são de extrema importância, pois as estruturas deles contêm as informações vitais ao cotidiano empresarial, por exemplo: dos dados pessoais ou financeiros que muitas vezes são confidenciais.

Tais dados possuem um sistema efetivo de backup, gerando cópias de segurança que permitem a recuperação de informações, caso ocorra algum problema.

Viu só como o DBA precisa estar apto ao mercado em que atuará? Dessa forma, ele pode atuar com segurança na administração de banco de dados.

Viu só! Um DBA precisa armazenar, organizar e proteger os dados por meio de um SGBD. A seguir, você verá este processo de administração de banco de dados com mais detalhes. Vamos lá?

Tarefas de um processo de administração de banco de dados

Um administrador de banco de dados precisa **disponibilizar a informação de forma rápida** aos colaboradores autorizados, para que tenham o acesso aos determinados níveis. Verifique agora uma sequência lógica do **processo**.



#Pracegover

Na imagem, um notebook aberto está sendo enquadrado e, enquanto as mãos de um programador estão trabalhando nele, uma está no teclado e a outra está no mouse. Na tela, há uma janela com um aviso de um programa sendo carregado.

- **Instalação do SGBD:** para que as informações estejam disponíveis por meio de um Sistema Gerenciador de Banco de Dados, o administrador deverá instalar e configurar cada item dos equipamentos envolvidos na execução do SGBD para ele ser utilizado por uma organização.

- **Efetivar a validação do modelo**

lógico: para que um sistema se desenvolva e funcione satisfatoriamente, é necessário realizar uma análise processual visando a concretização de uma série de tarefas – que devem ser desempenhadas na organização. Para maior entendimento e projeção dessas tarefas, é preciso conversar e entrevistar usuários para gerar o desenvolvimento de um modelo lógico de dados – que são baseados nas informações vindas de pesquisas e passarão, por fim, para a criação dos modelos físicos de dados.



#Pracegover

Na imagem, há uma série de papéis com gráficos impressos empilhados. Em cima deles, há uma caneta e uma lupa.



#Pracegover

Na imagem com fundo escuro, há uma projeção de um círculo claro com um ícone escrito "backup" no centro desse. Ao encontro dele, há um dedo indicador prestes a tocá-lo.

- **Sistema de backup:** a hipótese de informações perdidas, eventualmente, deve ser considerada, e por isso o sistema de backup garante o acesso dos usuários aos dados. A possibilidade de erros de comando, problemas de equipamento ou, ainda, problemas externos, como catástrofes naturais, devem ser consideradas. Desta forma, faz parte do planejamento do administrador de banco de dados a realização da cópia e do armazenamento dos dados em um local seguro.

- **Determinar alocação de espaço:** o administrador de banco de dados possui a tarefa de prover espaços livres no sistema, para que as tarefas dos usuários sejam bem executadas. Com isso, ele precisa entender o funcionamento para a alocação de espaço do programa SGBD e estar atento ao crescimento do banco de dados. Assim ele antecipa e evita um possível colapso do sistema por razão do excesso de dados que podem lotar o disco.



#Pracegover

Na imagem, sendo vista de cima, duas mãos digitam em um notebook aberto enquanto existem várias pastas de arquivo atrás dele.



#Pracegover

Na imagem, há uma janela projetada no centro com as instruções "username" e "password", junto com os botões "login" e "cancel".

- **Cadastramento de usuários autorizados:** para os usuários utilizarem um banco de dados, eles precisam ser cadastrados. Em um SGBD, ele necessitará fazer as solicitações por meio de seu cadastro. É responsabilidade do administrador de banco de dados prover este cadastro dos usuários.

- **Privilégios de acesso:** não basta o usuário apenas estar cadastrado em um sistema de banco de dados para acessar determinadas informações, ele precisa de uma autorização. Desse modo, garante-se a segurança dos dados em uma organização. Isso porque um banco de dados estará provido de informações dos mais diversos setores, e algumas dessas informações certamente serão de caráter sigiloso.



#Pracegover

Na imagem, há uma projeção de uma tela de computador com duas janelas e uma caixa de edição embaixo delas. Na janela da esquerda, há um escudo com vários códigos binários escritos. De maneira sobreposta, uma mão de um programador está levantada, enquanto uma outra tela de computador está ao fundo desfocada.



#Pracegover

Na imagem, há uma mão de um funcionário sendo enquadrada sobre um tablet enquanto interage com o aparelho. Embaixo dele, existem papéis que estão em cima de uma mesa de escritório. De modo sobreposto, existem vários gráficos no meio da imagem.

- **Entender e determinar a arquitetura do banco de dados:** o administrador de banco de dados precisa estar atento ao bom desempenho do banco que foi implementado, pois este contribui para a execução diária das tarefas dos colaboradores. Portanto, o administrador precisa constantemente estudar e entender a forma como o banco de dados escolhido atua, para que seja possível uma configuração rápida, bem como a resolução ágil de problemas que podem surgir com o tempo.

- **Observar o bom desempenho, segurança e estrutura do banco de dados:** um banco de dados pode, com o passar do tempo, receber um número grande de dados, ou ainda passar por modificações advindas de demandas externas do mercado. As mudanças são inevitáveis e, para que elas não prejudiquem o desempenho de um banco de dados, o administrador precisa estar atento às evoluções e às novas direções, sempre buscando informação e boas práticas que lhe permitam reestruturar o banco de dados nas necessidades da organização.



#Pracegover

Na imagem, há uma mão interagindo com o tablet sendo enquadrada. De modo sobreposto, há uma série de ícones espalhados pela tela com um cadeado e uma caixa de texto. Ao fundo, há um notebook aberto com várias linhas de código.

Apesar de um DBA ser o responsável por essas tarefas, ele poder ser confundido com outros profissionais que têm funções parecidas. Confira a seguir algumas carreiras semelhantes à de um administrador de banco de dados.

Carreiras em banco de dados próximas do DBA

Existem algumas carreiras que por vezes se confundem com as atribuições de um DBA, seja porque esses profissionais atuam muito próximos ou porque possuem algumas convergências. Apresentamos a tabela a seguir com algumas dessas características.

Carreira	Atribuições
Analista de Dados	Deve analisar os dados não estruturados e 'garimpar' os que forem relevantes, interpretá-los e fazer uso desses dados em estratégias de desenvolvimento de negócios.
Analista de <i>Business Intelligence</i>	Devem converter os dados em insights de valor estratégico e comercial, por meio de análises de dados aliadas à visualização de dados junto a técnicas de modelagem de dados. Analistas de BI devem identificar as tendências para auxiliar em tomadas de decisão de negócios em uma organização.
Analista Programador	É responsável pela elaboração dos requisitos de programação de sistemas, e com isso deve desenvolver, testar e implementar sistemas. Também concebe, coordena e orienta atividades de ordem de correção e manutenção de programas. Realiza codificação e testes, cria a documentação e produz manuais de operação.
Administrador de Dados	Atua como suporte à área de negócio nos aspectos lógicos da gestão de dados. A administração de dados abrange todo o sistema de informação, e não apenas os subsistemas automáticos de processamento de dados. Dessa forma, o objetivo do administrador de dados é a estratégia de negócio e seus reflexos nos requisitos de informação, e não a tecnologia em si.

Dentre estas carreiras, a que mais gera confusão, até por conta do nome, é a do administrador de dados. Veja, agora, no que ela se difere da carreira do administrador do banco de dados.

Importância dos dados na gestão

A importância dos dados na **gestão da informação** coloca o administrador de dados como uma figura cada vez mais presente no cenário das **organizações**. Desse modo, é importante diferenciarmos as atribuições do **administrador de dados**, para que não seja confundido com o **administrador de banco de dados (DBA)**, uma vez que **as duas carreiras lidam com dados e informações das empresas**.

O administrador de banco de dados surgiu antes no mercado, e, na medida em que a gestão de dados se posicionou de forma estratégica no mundo dos negócios, trouxe a parte da **administração de dados**. Confira, abaixo, as **principais tarefas** dela:

Atualização de modelos

Manter os modelos de uma corporação atualizados.

Interpretação de dados

Interpretar os dados e convertê-los em estratégias de mercado.

Qualidade das informações

Garantir a qualidade das informações e avaliar seu uso como solução corporativa para a empresa.

Fique atento! O principal objetivo do administrador de banco de dados está na **manutenção estratégica de negócios** e seus **impactos organizacionais**, não na tecnologia em si.

Assim sendo, a **administração de banco de dados** fornece suporte para a administração de dados, orientada para os cenários técnicos e físicos em gestão de dados. Enquanto isso, a **administração de dados** atua como apoio a tomadas de decisão de negócios, direcionada aos aspectos lógicos da gestão de dados.



Saiba mais

Para que você possa se aprofundar um pouco mais sobre a gestão de dados, atribuição voltada ao administrador de dados, convidamos você a assistir ao vídeo [gestão de dados](#).

A seguir, você entenderá mais detalhadamente sobre o **banco de dados**. Confira!

Bancos de dados

Anteriormente, você viu não só a relevância que os bancos de dados possuem no contexto da gestão de informação, como também a distinção entre o papel do administrador de dados e do administrador de banco de dados.

Agora chegou a hora de se aprofundar em relação ao conceito de **banco de dados**. Acompanhe com atenção!



Vídeo

Confira o [vídeo](#) sobre banco de dados.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Os bancos de dados podem ser utilizados em diversas situações e implantados em diferentes setores, independentemente do tamanho da empresa. Eles abrangem o gerenciamento das informações de recursos humanos, incluindo o armazenamento e a gestão de informação de caráter financeiro, logístico ou relacional.

As peculiaridades de cada modelo de negócio determinam o sistema de gestão de banco de dados ideal para atender às suas demandas. Com isso, além de estarem presentes desde os pequenos comércios até as grandes corporações, em sites, em blogs e até nas redes sociais. E esse sistema de banco de dados consiste na organização e armazenamento de dados e informações em um domínio específico. Ou seja, um agrupamento de dados que abordam um mesmo assunto destinado a funcionalidade similares e, por isso, necessitam de armazenamento para utilização contínua.

Vamos a um exemplo!

Imagine que você possui um modelo de negócio e construiu um *website*, onde pretende que os clientes acessem seus produtos, preencham cadastros e deixem seus dados. Para isso, tanto os produtos quanto as informações dos clientes precisam estar armazenados de forma eficiente.

Esses dados, se estiverem organizados corretamente, melhoram a performance e o trabalho das áreas de Tecnologia da Informação da empresa; sendo que seus profissionais poderão ter o acesso aos dados.

Por isso, a adoção de um bom sistema de banco de dados traz diversos benefícios para uma organização, impactando de forma positiva na produtividade das equipes e, consequentemente, nos resultados alcançados.

Confira, agora, os diferentes tipos de bancos de dados e como a escolha de qual será o mais adequado deve estar em concordância com a maneira como a empresa o utilizará: o primeiro deles é o banco de dados relacional, que era o formato mais dominante. Os bancos desse tipo se organizam via conjuntos de tabelas estruturadas em colunas e linhas. Ele provê uma forma muito eficiente e de alta flexibilidade no acesso às informações estruturadas.

Já o banco de dados orientado a objetos dispõe as informações em forma de objetos, de uma forma similar à programação orientada a objetos.

Os bancos de dados distribuídos consistem em dois ou mais arquivos alocados em diversos sites, em diferentes URLs. Eles podem ser armazenados em diversos computadores e estarem localizados fisicamente no mesmo local ou em redes dispersas.

Agora vamos aos *Data Warehouses*, que são destinados a prover consultas e análises em um repositório central de dados de forma rápida.

Há também o banco de dados NoSQL de caráter não relacional. Ou seja, esse banco de dados armazena e manuseia dados não estruturados e dados semiestruturados. Diferente de um banco relacional, que apresenta definições de como todos os dados inseridos em um banco devem ser organizados.

A força desse tipo de banco de dados NoSQL cresceu com o surgimento e popularização dos aplicativos na web e por isso, seus processos de complexidade aumentaram.

Outro tipo é o banco de dados gráficos, especializado em armazenar dados em bases de entidades, assim como os relacionamentos entre elas.

O banco de dados OLTP tem o caráter de armazenamento de dados rápido e analítico, construído para suportar grandes quantidades de transações efetuadas por diversos usuários.

Depois dele, temos o banco de documentos, que armazena dados em documentos similares aos objetos JSON (*JavaScript Object Notation*). Por conta disso, eles possuem poderosas linguagens destinadas à consulta. São indicados para os dados de documentos e para uso em geral. Eles, também, possuem alta escalabilidade horizontal, que acomoda grandes volumes de dados. Um exemplo dele é o MongoDB, que é o banco de dados NoSQL mais popular.

Por último, temos a Chave-Valor, que possui um estilo mais “simples” de banco de dados, onde cada item contém chaves e valores. Os valores podem ser de qualquer espécie de dados, como um texto, um número, um JSON. E eles podem ser recuperados indicando a sua chave, o que torna a consulta bastante simples. Esse tipo é recomendado para armazenamento de grandes quantidades de dados, mas sem a necessidade de consultas complexas. Um exemplo de seu uso corriqueiro é o de armazenamento de dados em cache. Redis e DynamoDB são os bancos mais populares dessa modalidade.

Agora que você já conhece o conceito e os diferentes tipos de banco de dados, continue seus estudos com as outras modelagens de banco de dados existentes.

Após assistir ao vídeo, você observou não só o conceito de banco de dados, como também os seus diferentes tipos: os relacionais, os orientados a objetos, os distribuídos, os *Data Warehouses*, os NoSQL, os gráficos, os de documentos e os Chave-Valor.

Todos estes bancos de dados são de **uso no mercado**. Contudo, existem ainda outras modelagens menos utilizadas, adaptadas para os ambientes científicos, financeiros ou áreas mais específicas.

Devido aos avanços tecnológicos, outras modalidades se impõem por sua novidade, como os bancos em nuvem para processos de automação e *machine learning*, os quais levam a gestão de dados para novas esferas e direções.

Destacamos, a seguir, alguns bancos de modelagem recentes:



#Pracegover

Na imagem, há uma fechadura no centro e, em volta dela, existem vários códigos binários com cores claras e com um fundo escuro.

- **Bancos de dados de código aberto:** esse sistema pode ser de dados SQL ou NoSQL e de código-fonte aberto.



- **Bancos de dados em nuvem:** em enorme crescimento no mercado, trata-se de um acervo de dados estruturados ou não estruturados, alocados em uma plataforma de armazenagem em nuvem privada, pública ou híbrida. Dois estilos desse tipo de banco se destacam: o tradicional e o banco de dados como serviço (DBaaS). No caso DBaaS, as tarefas administrativas e manutenção são executadas por meio de provedores de serviços.

#Pracegover

Na imagem, há três nuvens formadas por circuitos elétricos e há uma série de códigos binários claros em um fundo escuro.



#Pracegover

Na imagem, há uma série de linhas de códigos em uma tela.

- **Banco de dados de documentos/JSON:** tipo de banco projetado para armazenamento, recuperação e gerenciamento de informações com orientação aos documentos. Os bancos de dados de documentos constituem uma forma moderna de alocação de dados em formato JSON, em contraponto ao tradicional sistema de linhas e colunas.



#Pracegover

Na imagem, há um notebook aberto e, de modo sobreposto, existem várias linhas de código e, no meio, há a palavra JSON escrita.



#Pracegover

Na imagem, existem quatro ícones: três rodas dentadas, um rosto de perfil com rodas dentadas dentro dele, um cérebro humano e um globo. Em um fundo escuro, existe um grande circuito elétrico e, no lado esquerdo, há uma mão com o dedo indicador ereto.

- **Bancos de dados autônomos:** são bancos de dados independentes e inovadores, com base em nuvem e integração, com *machine learning* (que aprende, automatiza e ajusta o trânsito no banco de dados). Provê mais segurança, backups rápidos e prontas atualizações a outras tarefas gerenciais de rotina, que em moldes tradicionais são executadas por administradores de banco de dados.

Notou como o banco de dados está presente em diversos lugares? Acompanhe, a seguir, a linguagem de pesquisa declarativa padrão para o banco de dados relacional, o SQL (*Structured Query Language*)!

SQL

Ao trabalhar com bancos de dados, você se envolve, de uma forma ou outra, com o **SQL (Structured Query Language)**, presente na vida de **desenvolvedores e profissionais de DBA**. O seu conhecimento é **requisitado no mercado** em virtude de uma parcela expressiva do bancos de dados estarem estruturados em SQL e em seus formatos de implementação, como o T-SQL (no caso do *SQL Server* da Microsoft) ou o PL/SQL de patente da Oracle.



#Pracegover

Na imagem, existem quatro ícones: um com três rodas dentadas, o outro com um notebook aberto, o outro com servidor, e o último com a palavra SQL, o qual está sendo tocado por um homem de terno.

Considerado um dos recursos para administração de banco de dados **mais utilizados do planeta**, o SQL está ligado aos **bancos de dados relacionais**. Dessa forma, apesar de não ser a única maneira de estruturar um banco de dados, certamente, o SQL é o mais conhecido e utilizado. Como tal, ele está presente em relações de acesso, consultas e alteração de dados estruturais em bancos de dados de uma organização. Ademais, é uma ponte de comunicação entre DBAs e desenvolvedores.

Nesse conteúdo você verá como funciona o primeiro **acesso ao SQL**.

Os sistemas baseados em SQL são utilizados na **realização de tarefas**, como programar as rotinas, *queries*, realizar consultas ou manipular dados. Inclusive realizam as tarefas simples em estruturas de tabelas e estão fundamentados em quatro funcionalidades: *insert* (inserir), *delete* (excluir), *update* (atualizar) e *search* (pesquisar).

Essa ferramenta é **altamente versátil**, permitindo que se lide diretamente com os dados e interaja com softwares e diversos sistemas digitais. As sintaxes que compõem o SQL possuem algumas subdivisões. São elas:

DML – *Data Manipulation Language*

São comandos de alteração das informações contidas em tabelas, o que pode ser para a inserção ou exclusão de dados (*select*, *delete* e *insert*).

DDL – *Data Definition Language*

São comandos de modificação do banco de dados (*drop* – apaga um objeto, *create* – cria um objeto).

DCL – *Data Control Language*

Atua nas permissões, restrições e bloqueios no banco de dados (*grant* – permite acesso ou, ainda, modificações no banco de dados).

DTL – *Linguagem de Transição de Dados*

Responsável por registrar e salvar quaisquer alterações realizadas por usuários (*commit* – autoriza o salvamento de eventuais alterações).

Apesar das novidades da tecnologia de bancos de dados, como por exemplo a nuvem (*SQL Cloud*), a estrutura básica do SQL, usada há mais de 40 anos, se mantém relevante e popular. Justamente por passar por poucas modificações, é **universal e estável**, sendo ainda de altíssima relevância no trabalho de um DBA.

O SQL e seus componentes

Depois de você estudar mais sobre o tipo de banco de dados SQL, bem como as suas características e singularidades, neste tópico, você entrará em contato com os seus respectivos **componentes** que estão divididos da seguinte maneira:

Database Engine

O serviço principal de um banco de dados SQL Server, possuindo várias *features*, como replicação, *full-text*, *service broker* etc.

Analysis Services

Processamento analítico, cruzamento de dados, geração de relatórios, muito usado em BI (*Business Intelligence*).

Reporting Services

Desenvolvimento de relatórios e geração de gráficos.

Integration Services

Tarefas de ETL, integração, importação e exportação de dados.

Aqui você viu os componentes do SQL com as suas principais características. Todavia é importante que você também conheça outro aspecto dele: as ferramentas de gestão.

Ferramentas de gestão do SQL

Agora, chegou o momento de você entender quais são as **ferramentas de gestão do SQL**.

SQL Server Management Studio

Principal ferramenta de gestão do SQL.

SQL Server Configuration Manager

Ferramenta para manutenção no nível do banco de dados. Por exemplo, trocar a porta de serviços, reiniciar ou parar o banco de dados, trocar parâmetros de inicialização etc.

SQL Server Profile

Permite a realização de consultas em tempo real ao banco de dados.

Saiba mais



Muitas empresas utilizam simultaneamente os bancos SQL e NoSQL, que, integrados, podem combinar, por exemplo, a especificidade de dados provenientes de usuários em aplicações web utilizados em processos de ***business intelligence***.



Mas você deve estar se perguntando: o que é possível realizar a partir desta integração?

É possível efetuar uma importação dos dados de um banco não relacional e, após isso, tratá-los e armazená-los em tabelas para avaliação em outra oportunidade. E o inverso também é possível.

Por isso, as empresas gigantes no mercado, como Amazon ou Apple, utilizam muitas aplicações NoSQL integradas com bancos relacionais SQL.

Para que você compreenda melhor, a tabela a seguir ilustra as diferenças entre ambos. Observe com atenção!

	SQL	NoSQL
Modo de armazenagem	Tabelas de colunas e linhas fixas.	Documentos do tipo JSON, Chave-Valor etc.
Histórico	Criado na década de 1970, com foco em redução de dados duplicados.	Desenvolvido em 2000 com objetivos de escalabilidade e rápida mudança de desenvolvimento.
Exemplos	Oracle, MySQL, Microsoft SQL Server, e PostgreSQL	Documento: MongoDB e CouchDB, Chave-Valor: Redis e DynamoDB, Wide-column: Cassandra e HBase, Graph: Neo4j e Amazon Neptune
Esquemas	Rígidos	Flexíveis
Escalonamento	Vertical (mais poder de processamento em máquina única).	Horizontal (distribuído em duas ou mais máquinas).
Transações	Suportado	A maioria não suporta, o MongoDB sim.
Joins	Normalmente necessário	Normalmente não necessário
Mapeamento de Dado para Objeto	Requer um ORM (<i>Object-Relational Mapping</i>)	Pode não precisar de um ORM. Os documentos no MongoDB mapeiam diretamente para dados de estrutura das maiorias das linguagens.

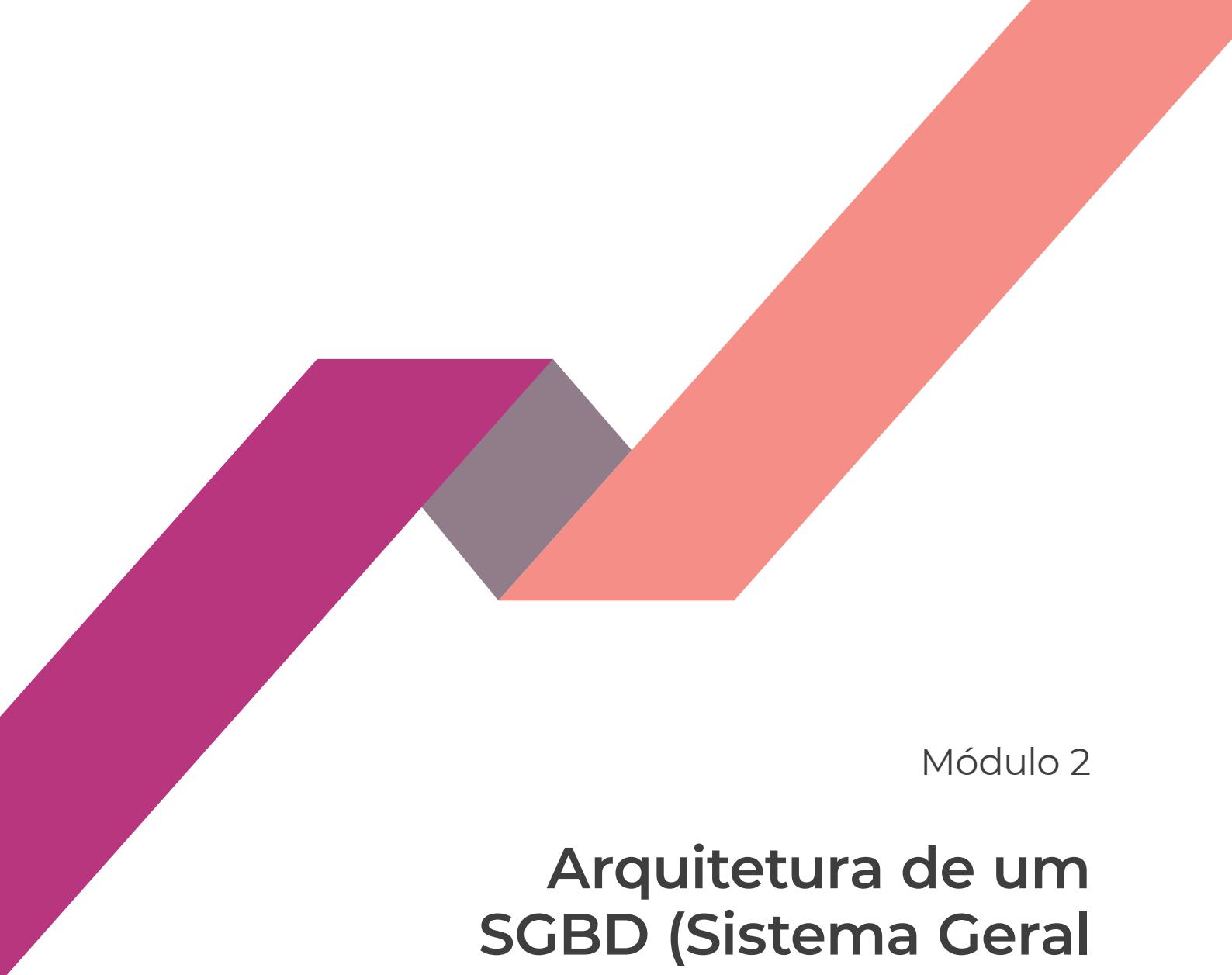
Parabéns! Você chegou ao final do Módulo 1!

Até aqui, você compreendeu o papel do administrador de banco de dados, o contexto profissional e mercado de trabalho em que atua; as tarefas de um processo de administração de banco de dados; as carreiras relacionadas a banco de dados com funções às desempenhadas pelo DBA; e a importância dos dados na gestão de informação das empresas.

Além disso, você também teve acesso ao conteúdo mais técnico, principalmente em relação ao conceito de banco de dados e seus diferentes tipos, bem como a linguagem de pesquisa declarativa padrão para o banco de dados relacional, o SQL, e seus componentes e ferramentas de gestão.

No próximo módulo, continuaremos a estudar o conteúdo mais técnico, sobretudo, relacionado à arquitetura de um Sistema Geral de Banco de Dados, o SGBD.

Preparado? Então vamos lá!



Módulo 2

Arquitetura de um SGBD (Sistema Geral de Banco de Dados)

Arquitetura de um SGBD (Sistema Geral de Banco de Dados)

Neste segundo módulo, você compreenderá as estruturações física e lógica de um banco de dados e um pouco sobre a maneira como um Sistema Geral de Banco de Dados é configurado.

Dessa forma, percorrerá diversos pontos importantes, como: os modelos de dados, esquemas e instâncias (que são pontos-chave para uma arquitetura eficaz de tal sistema). Confira!

Modelos de dados

Os modelos de dados são utilizados na descrição e na construção das estruturas lógica e física em um banco de dados. Os tipos de dados que têm relacionamentos entre eles e suas restrições são conhecidos como estrutura ou nível, que se dividem basicamente em duas modalidades. Vamos conhecê-las!

Alto nível

Denominado modelo de dados conceituais ou modelo Entidade-Relacionamento. O seu principal conceito é uma projeção dos dados com a maior proximidade possível da visão que o usuário obtém deles.

Baixo nível

Nesse caso, é mais conhecido como um modelo de dados de caráter físico, pois ele fornece uma visão detalhada da maneira como os dados estão armazenados em um local.

Com esses conceitos de modelos de dados bem claros, é preciso destacar que existe ainda outro ponto relevante para a configuração dos sistemas gerais de banco de dados. Estamos falando dos “esquemas”, conceito que veremos no próximo tópico.

Esquemas (*schemas*)

No momento da configuração do SGBD, é imprescindível que exista uma representação da sua estrutura com as suas regras detalhadas, principalmente, porque tal sistema, na grande maioria dos casos, é feito por uma equipe de desenvolvedores.

Logo, para que isso ocorra de modo eficaz e otimizado, é necessário haver um esquema. Convidamos você a assistir ao vídeo sobre os **esquemas (*schemas*)**, para conhecer, em maiores detalhes, sua funcionalidade. Acompanhe com atenção!



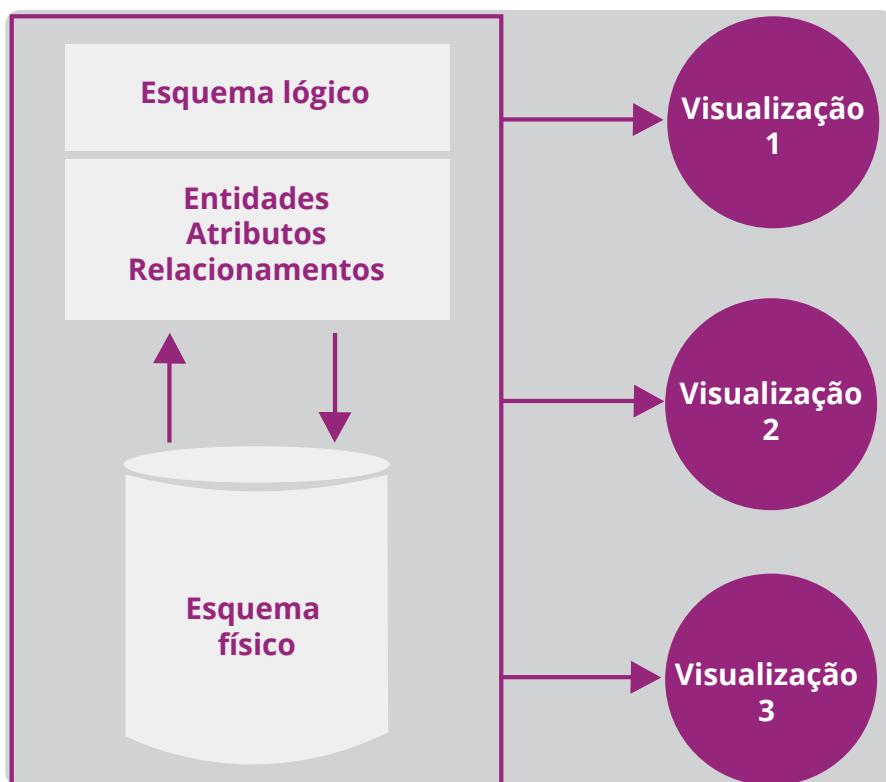
Vídeo

Confira o [vídeo](#) sobre esquema de banco de dados.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Para esse estudo, vamos estabelecer uma distinção entre a descrição do banco de dados e o próprio banco de dados.

A descrição é chamada de “Esquema de banco de dados”, e sua apresentação ocorre por um diagrama de esquema.



#Pracegover

Na imagem, retângulo em pé e, dentro dele, há duas caixas de texto interligadas por setas. Na primeira, está escrito logical schema, Entities e Attributes Relationships e, na segunda, está escrito physical schema. Na direita, saindo do retângulo em pé, partem três setas em direção a três círculos, nos quais estão escritos view 1, view 2 e view 3.

Um esquema se refere a uma representação visual de uma estrutura de banco de dados, a um conjunto de regras que abrange um banco de dados, ou o composto de objetos relacionados a um usuário específico.

A representação de um esquema se refere à totalidade, ou a uma parte da base relacional de dados, considerando desde a representação visual até um composto de fórmulas, que são chamadas restrições de integridade que controlam um banco de dados.

Essas fórmulas são representadas em forma de linguagens de demonstração de esquemas de dados, como o SQL.

Um esquema de banco de dados aponta como componentes de composição de um banco de dados, que se relacionam mutuamente, incluindo as tabelas, projeções, exibições, rotinas e procedimentos, eventualmente armazenados, e diversos outros elementos. São imagens “esquemáticas” que apresentam uma noção de estrutura de um banco de dados, ou de suas partes.

A partir desse estudo, você pôde compreender as principais características de um esquema de banco de dados e verificar como ele poderá auxiliar o desenvolvedor.

No vídeo, você acompanhou a explicação das principais características de um esquema dentro do contexto de SGBD, considerando tanto a sua função quanto as fórmulas que o compõem.

Inclusive, vale ressaltar que a criação de um esquema de banco de dados auxilia os desenvolvedores a compreenderem, conceitualmente, **como um software interage com o banco de dados**. Esse processo de concepção de um esquema se chama **modelagem de dados**, e existem **dois estilos** mais utilizados para os esquemas de banco de dados.

Esquema de banco de dados lógico

Abrange restrições de caráter lógico aplicadas aos dados armazenados. Pode, entre outras definições, cobrir as restrições de integridade, exibições e tabelas.

Esquema de banco de dados físico

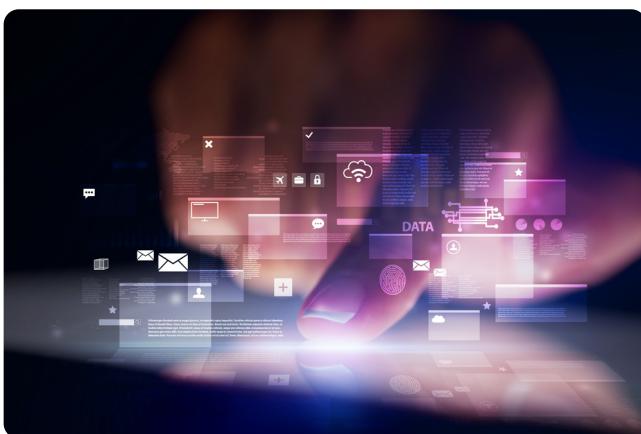
Contempla a forma de armazenagem física no sistema de armazenamento, incluindo os arquivos e índices.

Os esquemas de banco de dados indicam as tabelas ou relações entre os dados que compõem o banco; sobre quais campos serão inseridos em cada tabela; e as suas restrições.

O conceito de esquema é muito relevante para a configuração de um SGBD. No entanto, para que isso ocorra, é imprescindível que você entenda e domine outro elemento importante: as instâncias. Para entendê-las melhor, acompanhe o próximo tópico, para entender melhor sobre elas.

Instâncias

As instâncias são os **registros em um banco de dados**, estruturas de memória que formam a área de **memória compartilhada**, denominada de **área global do sistema (SGA – System Global Area)**. Essas instâncias atuam em processos de segundo plano (*background processes*). Os processos rodam sempre no servidor, localizados na memória RAM e no Processador (CPU). A instância pode existir de forma independente dos arquivos em um banco de dados.



Para um maior entendimento, imaginemos um processo simples de uma *startup* em um banco de dados **Oracle**.

#Pracegover

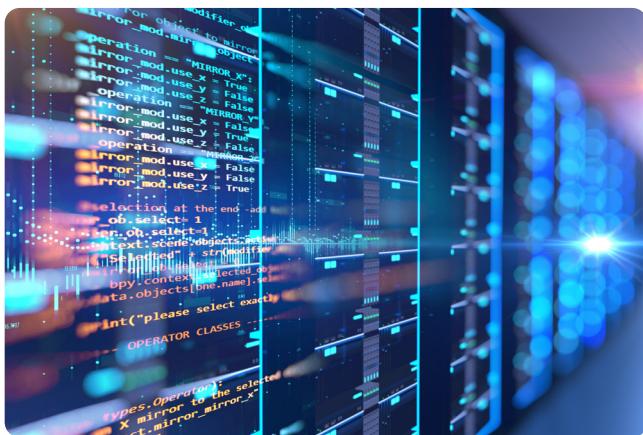
Na imagem, vários retângulos com transparência indicam janelas de aplicativos de um smartphone. No fundo uma mão mexe na tela.

Nesse momento, uma instância realiza uma busca procurando um arquivo de parâmetros (SPFILE ou PFILE), uma espécie de guia onde as regras de configuração estão contidas (memórias, localização de arquivos etc.).



#Pracegover

Na imagem, duas mãos digitam em um notebook e na frente tem uma representação de arquivos interligados que saem de um arquivo maior.



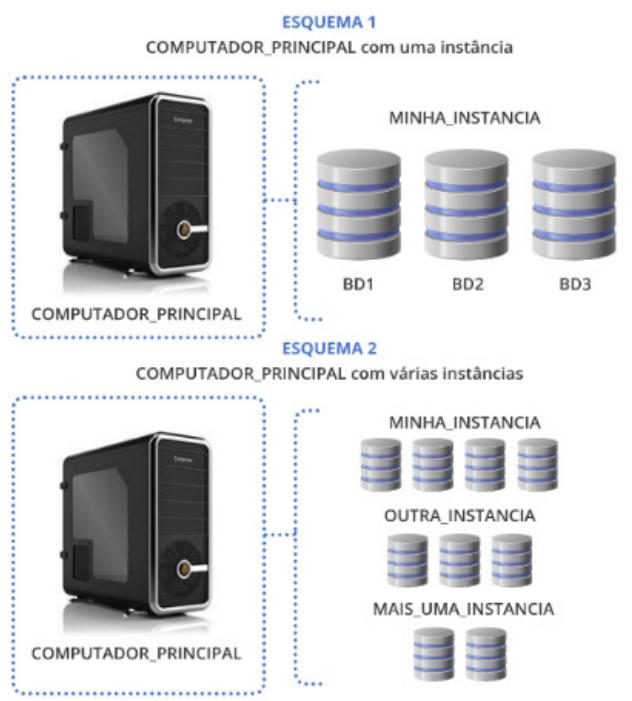
#Pracegover

Na imagem há uma fileira de bancos de dados físicos com códigos escritos neles.

E, uma vez identificadas as regras de parâmetro, as instâncias procedem com a montagem e o acesso ao banco de dados.

As instâncias acompanham as mudanças realizadas na base do banco de dados. Com isso, o SGBD deve garantir que as instâncias atendam ao esquema do banco de dados, acatando o que foi definido em sua estrutura e restrições.

Para saber mais sobre a sua aplicação, observe as informações na imagem abaixo:



Ao instalar o SQL em um computador, ele se transforma em servidor, tornando-se o computador principal de uma rede. Esse computador possui uma instância de computador principal, que poderia estar ligada a outras instâncias, dependendo da natureza dos dados. E é muito interessante que as instâncias estejam distribuídas em locais diferentes.

#PraCegoVer

Na parte superior da imagem há um computador com uma instância. E na parte inferior, um computador com várias instâncias.

Note que a estrutura de um SGBD tem o objetivo de separar as aplicações de usuário dos dados físicos, como apresentamos a seguir:

Nível/esquema interno

Utiliza uma modelagem de dados que mostra a estrutura de armazenamento físico, o detalhamento dos dados armazenados e as vias de acesso no banco de dados.

Nível/esquema conceitual

Descreve a estrutura do banco de dados em sua totalidade, mas sem o detalhamento dos dados armazenados, apresentando de forma genérica.

Nível/esquema de visão

Descreve no banco de dados quais e quantos usuários possuem acesso ao banco.

A **arquitetura de um banco de dados**, ao ser projetada, abrange um conjunto de regras e modelagens, que governam os dados, e **determina como os dados coletados devem ser armazenados, organizados e integrados na organização**. Dessa forma, acompanha a gestão de dados, que define os padrões para os sistemas coletores e armazenadores de dados, com base na visão de modelo e funcionamento das interações entre os sistemas.

Uma boa arquitetura de dados: define os padrões para os sistemas; descreve as estruturas de dados utilizadas pela empresa; aborda os dados em armazenamento, os dados em uso e em movimento; aponta as descrições de grupos de dados, os itens e o seu mapeamento, para o controle de qualidade da gestão dos metadados.

Criar um banco de dados não é apenas pensar nas estruturas física e lógica, ele deve apresentar uma **conexão com as decisões estratégicas de negócio**. A gestão de dados é um forte diferencial competitivo.

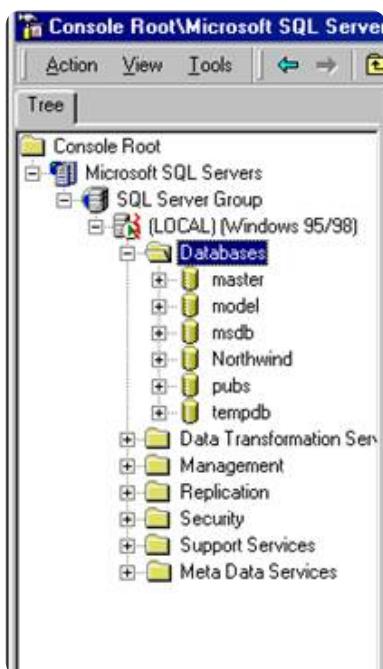


Saiba mais

Caso queira saber mais sobre a gestão de dados, indicamos o vídeo [análise de dados para gestão de crise](#). Vale a pena conferir!

Explorando um banco de dados relacional

No momento de criação de um banco com base em *SQL Server*, o mecanismo de instalação cria os **bancos de armazenagem de dados nativos** do sistema. Esses bancos, chamados de **Bancos de Dados do Sistema**, são utilizados com **finalidades internas**. Ou seja, um **Banco de Dados Relacional** armazena e fornece acesso a pontos de dados relacionados entre si a partir de tabelas. Observe como se organiza sua estrutura.



#PraCegoVer

Na imagem, há uma captura de tela mostrando uma janela que contém banco de dados de um sistema. Na parte superior, está a barra de ferramentas gerais. Embaixo, está o painel que expõe o banco de dados de um sistema.

No próximo *podcast*, você conhecerá quais são seus **componentes**. Acompanhe!



Podcast

Confira o *podcast* com os componentes do banco de dados.

Perdeu algum detalhe? Confira o que foi abordado no *podcast*.

Olá! Agora que você já sabe como criar e instalar o mecanismo que gera um banco de dados, vamos conhecer seus componentes?

O primeiro deles é o **Master, trata-se do banco de dados ‘mestre’**. Ele armazena as informações de metadados de toda a instância, as configurações do servidor, as informações sobre todos os bancos da instância e as informações de inicialização. Além de ter todas as informações de nível de sistema registradas em uma instância do SQL Server.

O **Model**, por sua vez, é um guia ou um **modelo** para os bancos de dados que serão criados. Digamos que cada banco criado será uma cópia do *Model*, que proporciona as modificações do tipo tamanho, agrupamento, modelo de recuperação e outras opções. Uma vez que as alterações são definidas, elas serão aplicadas a qualquer banco de dados criados na sequência.

O **MSDB** é o banco que armazena o serviço SQL Agent e o local onde são **armazenados os dados**. Esse banco é utilizado pelo SQL Server Agent para programação de *alerts* e *jobs*.

Já o **Northwind** é o banco de dados de exemplo para **treinamento**. Ele necessita do download de extensões para ser ativado.

O **Pubs** é o banco de dados do tipo **somente leitura**, ele contém objetos de sistema *inclusis* no SQL Server. Estes são fisicamente constantes no banco de dados Recurso, mas representados logicamente no esquema sys do banco de dados.

Por fim, o **TempDb** é a área de **memória** para a retenção de objetos temporários ou conjuntos de resultados de caráter intermediário. Nele onde são armazenados os dados temporários.

Agora que você adquiriu um pouco mais de conhecimento em relação aos componentes dos bancos de dados, confira no próximo tópico o que são as tabelas do sistema, outro aspecto muito importante na configuração de um SGBD.

Neste tópico, você viu sobre a criação e instalação do mecanismo que gera bancos de dados no sistema, bem como as suas finalidades. Além disso, no *podcast*, foi possível entender os seus componentes. Isso tudo possibilitou que você adquirisse mais experiência em relação a eles.

Mas fique ligado! Ainda é necessário conhecer outro aspecto muito relevante desse contexto: as chamadas tabelas do sistema, as quais veremos no próximo tópico.

Tabelas do sistema

Na sua criação, o SQL Server elabora as tabelas do sistema, incluindo as tabelas de forma automática. É preciso atenção com as tabelas de sistema: elas **não devem ser alteradas diretamente por nenhum usuário**, nem devem sofrer tentativas de modificação com alguma instrução definida, como: *delete*, *update*, *insert* ou qualquer outro comando.

No exemplo a seguir, apresentamos uma tabela de sistema e seus modos de exibição.

Tabelas do sistema	Exibições ou funções de sistema	Tipos de Exibição
SYSALTFILES	sys.master_files	Exibição de catálogo
SYSCACHEOBJECTS	sys.dm_exec_cached_plans sys.dm_exec_plan_attributes (Transact-SQL) sys.dm_exec_sql_text sys.dm_exec_cached_plan_dependent_objects	Exibição de gerenciamento dinâmico
SYSCHARSETS	sys.syscharsets	Exibição de compatibilidade

De acordo com o que você observou na última coluna e seus três termos, para que entenda melhor cada significado, especificamos de forma mais detalhada cada um deles a seguir.

Exibição de catálogo

Um **conjunto de arquivos de armazenagem de parâmetros** da linguagem DDL (*Data Definition Language*) ou Linguagem de Definição de Dados. Fornece o acesso aos metadados contidos no servidor. A partir desse catálogo, é possível determinar os registros alterados no banco de dados durante um período, por exemplo.

Exibição de gerenciamento dinâmico

Retornam os dados que **informam a situação do servidor**. Essas informações podem ser usadas no monitoramento de uma instância no servidor, realizar um diagnóstico de problemas, verificar a integridade da instância e acompanhar o desempenho do servidor.

Exibição de compatibilidade

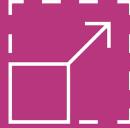
As tabelas contidas na exibição de compatibilidade possuem a tarefa de manter a **compatibilidade com as versões anteriores**.

Viu como as chamadas tabelas do sistema são importantes para a configuração do SGBD?

Só que, além disso, é de igual importância conhecer os tipos de arquivos que terão a função de compor os bancos de dados a serem criados. A seguir, você conhecerá mais sobre eles. Vamos lá!

Tipos de arquivos

Para a composição dos diferentes tipos de banco de dados é necessário que os arquivos sejam gerados no momento da elaboração do banco. Assim, neste tópico, você vai conhecer os tipos de arquivos de banco de dados. O SQL Server possui basicamente **três tipos de arquivos**, observe com atenção cada um deles:

Primário 	<p>Os arquivos de dados primário contêm informações responsáveis pela inicialização do banco de dados, e ainda indicam outros arquivos. Os dados de usuário e objetos podem estar contidos nesse arquivo ou ainda em arquivos de dados secundários. Os primários são de criação do usuário. O sistema faz uso do que foi indicado pelo desenvolvedor, para a criação de arquivos de extensão .mdf.</p>
Secundário 	<p>Os arquivos secundários definidos pelo usuário e costumam armazenar os seus dados. Podem ser utilizados na distribuição de dados entre os discos, inserindo cada arquivo em uma unidade de disco à parte. São utilizados para a expansão de um banco de dados, se ele ultrapassar seu tamanho inicial. São arquivos opcionais e sua extensão é .ndf.</p>
Logs de transações 	<p>Logs de transações são arquivos que armazenam informações de <i>log</i>, utilizadas na recuperação ou restauração de um banco de dados. Deve existir um arquivo de <i>log</i> a cada banco de dados minimamente. A extensão de nome de arquivo indicada para arquivos de transação é .ldf.</p>

É importante que você saiba que esses arquivos pertencem aos **grupos de arquivos**. Desse modo, torna-se mais fácil a busca por dados, além de melhorar o desempenho do banco de dados.

Veja que os nomes dos grupos são semelhantes aos nomes dos tipos de arquivos.

Grupo primário

Grupo de arquivos que **contém o arquivo primário**. Todas as **tabelas** do sistema são alocadas nesse grupo.

Grupo secundário

Qualquer **grupo de arquivos que seja definido pelo usuário**, quando cria ou modifica o banco de dados.

Grupo padrão

Grupo de objetos criados no banco de dados **sem especificação** de grupo de arquivos ao qual pertencem.

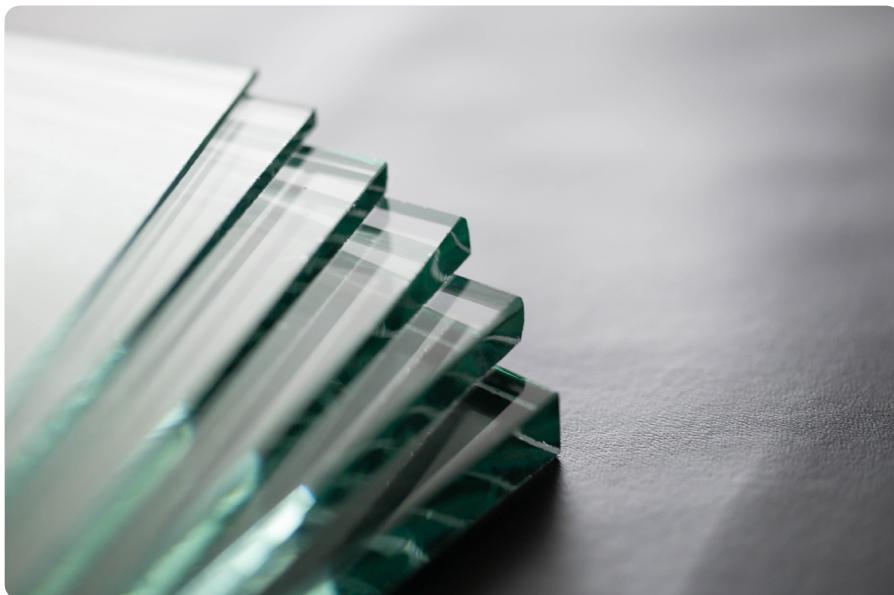
Lembre-se! Um **grupo de arquivos ou filegroup** contém um ou mais arquivos de um banco de dados, **formando uma unidade**. No entanto, esses arquivos de um grupo específico podem estar separados, ou seja, gravados em diferentes discos rígidos. Mesmo assim, um arquivo pode ser parte de um único grupo de arquivos.

Assim, depois de ver os principais conceitos, no próximo tópico, vamos conferir o passo a passo de como criar um banco de dados.

Criar um banco de dados

O **primeiro passo** para criar um banco de dados é a **análise de requisitos**, que auxiliará na compreensão das dimensões do banco de dados. Por isso, as informações devem ser coletadas em entrevistas com os usuários-chave.

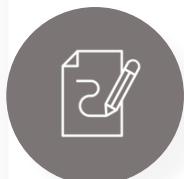
A seguir, veja um exemplo com os dados coletados em uma situação hipotética.



#Paracegover

Em cima de uma superfície, placas de vidro estão empilhadas no canto inferior esquerdo da imagem.

Exemplo



Imagine que a empresa trabalhe com embalagens de vidro de tamanhos e capacidades diferentes. Ela atende clientes de diversos perfis (empresas de cosméticos, farmacêuticas, alimentícias etc.), sua necessidade se concentra em processos de RH e possui 7.700 colaboradores.

Como primeira medida, um banco de dados de 15M será instaurado, com incremento de 15%, devido ao aumento constante da quantidade de funcionários.

Veja o processo de [criação de um banco de dados no SSMS](#).

Para criar um banco de dados no modo **T-SQL**, observe como a **sintaxe de código** deve ficar. Não esqueça de selecionar os destaques para ver as descrições e assim entender melhor seus comandos, parâmetros e funcionalidades:

```
USE Master
GO
CREATE DATABASE Nome _ do _ Banco _ de _ Dados
ON
( NAME = nome _ dat,
  FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\nome.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = nome _ log,
  FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\nomelog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB ) .
GO
```

Nome_do_Banco_de_Dados <i>CREATE DATABASE</i>	Indica o nome do banco de dados.
<i>ON</i>	Indica o caminho de armazenagem de arquivos.
<i>FILENAME</i>	Usado para definir um disco de armazenagem de arquivo.
<i>SIZE</i>	Determina o tamanho de um arquivo. Pode utilizar Kilobytes (KB), Megabytes (MB), Gigabytes (GB) ou Terabytes (TB).
<i>MAXSIZE</i>	Define o tamanho máximo de um arquivo. Deve ser informado, ou os arquivos crescerão sem controle e ocuparão toda a base de dados.

<i>FILEGROWTH</i>	Define o incremento de crescimento do arquivo. Pode ser em kilobytes, megabytes ou em porcentagem (%).
<i>LOGON</i>	Aponta de qual local no disco os arquivos <i>log</i> devem ser definidos.

Para mais detalhes, veja o material sobre [criação de um banco de dados no TSQL](#).

Agora, acompanhe a série de materiais, abaixo, com as demonstrações de **como administrar sua estrutura**, para que as informações futuras sejam apresentadas e edificadas:

-  [Como visualizar o tamanho de um banco de dados no SSMS](#)
-  [Como expandir um banco de dados no SSMS](#)
-  [Como adicionar um arquivo ao banco de dados no SSQl](#)
-  [Como reduzir um banco de dados no SSQl](#)

Saiba mais



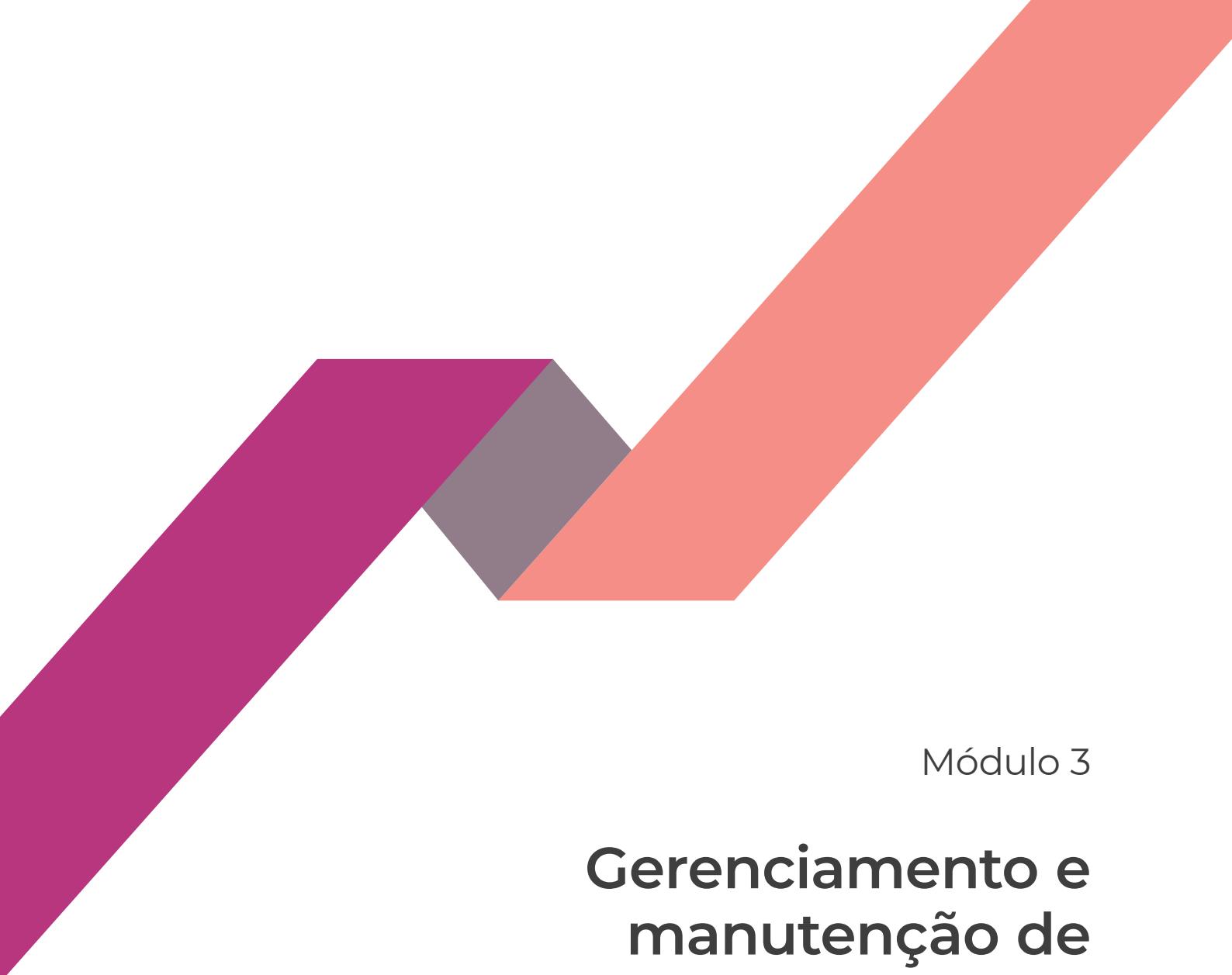
O desenvolvimento de **ferramentas visuais** para o banco de dados facilita a vida de administradores de banco de dados, pois seus recursos intuitivos, em muitos casos, dispensam o uso de codificação para que se estruture um BD. Mas existem outras opções, e com um pouco de pesquisa será possível encontrar diversas ferramentas interessantes com bons recursos visuais.

Para ajudá-lo nisso, indicamos o vídeo sobre o modo de [criar um projeto de banco de dados com Visual Studio 2017](#).

Parabéns! Você finalizou o Módulo 2!

Aqui, você adquiriu conhecimentos técnicos sobre o SGBD, principalmente, em relação à sua arquitetura. Por conta disso, estudou sobre os diferentes modelos de dados, os conceitos de esquemas, instâncias, bem como entendeu mais sobre o processo de criação de um banco de dados relacional.

No próximo módulo, estudaremos sobre da continuidade da criação dos sistemas de dados, isto é, o processo de manutenção e de gerenciamento deles. Até lá!



Módulo 3

Gerenciamento e manutenção de sistemas de dados

Gerenciamento e manutenção de sistemas de dados

Conforme você viu, não basta apenas criar um sistema de dados. É fundamental ir além disso, sobretudo, quando você estiver em seu futuro ambiente profissional.

Sendo assim, neste módulo, você entenderá sobre dois processos muito relevantes para os administradores de bancos de dados: as informações sobre scripts manuais para o gerenciamento e a manutenção de banco de dados.

Tudo certo? Então vamos começar com a manutenção de banco de dados.

Manutenção de banco de dados

O processo de criação de banco de dados é muito importante e igualmente trabalhoso, pois é necessário que tudo o que foi elaborado não se perca e que funcione adequadamente. Assim, é preciso que tal banco tenha uma **manutenção constante**, assunto que iremos tratar neste tópico.

Mesmo **após a instalação** exitosa de um banco de dados, ainda existem **tarefas preliminares** a realizar. Você, como um profissional de DBA, precisa realizar algumas **configurações**, que dizem respeito às seguintes tarefas:

Pontos de verificação manual do banco de dados

Validação de tabela

Grupo de arquivos

Para a boa realização dessas atividades, existem **comandos de gerenciamento do SQL Server**, que auxiliam esses processos, muitas vezes **manuais**. Vamos conhecer os três comandos mais utilizados?



CHECKPOINT

Esse comando gera um **ponto de verificação manual** no banco de dados no instante da execução.

Acompanhe neste material o passo a passo de [como aplicar o checkpoint](#).

#PraCegoVer

Imagen de uma mulher de óculos com um tablet em suas mãos, ao fundo diversos computadores com fundo azul.

DBCC (*Database Console Commands*)

Essas instruções agem como **Comandos de Console de Banco de Dados (DBCC)** para o SQL Server e são divididas em **quatro categorias**. Vamos conhecê-las:

Manutenção	Realiza tarefas de manutenção em um BD, índice ou grupo de arquivos.
Diversas	Habilitam sinalizadores de rastreamento, removem uma DLL da memória etc.
Informativas	Informam as estatísticas da consulta, exibem a situação dos sinalizadores de rastreamento.
Validação	Realiza validação em tabela, grupo de arquivo, entre outros meios.

Sintaxe:

```
DBCC SHRINKFILE  
(file_name/file_id, EMPTYFILE, target_size, NOTRUNCATE |  
TRUNCATEONLY)  
WITH NO_INFOMSGS
```

Veja aqui o [processo de aplicação do DBCC](#).

SP_SPACEUSED do TRANSACT-SQL

Exibe o número de linhas, o espaço reservado em um banco de dados, ou o espaço em disco usado por uma tabela, ou ainda, um *view* no banco de dados atual. O nome indica sua função: **SPACE USED → ESPAÇO USADO**.

Sintaxe:

```
sp_spaceused @objname = 'objname',
@updateusage = 'updateusage'
```

Saiba mais



A **criação de planos de manutenção**, embora seja uma funcionalidade existente no SQL Server desde 2000, **não é utilizada pelos DBAs**. Sabemos que o banco de dados possui um conjunto de tarefas que necessitam de execução de forma rotineira, decorrente da amplitude da base de dados. Desta forma, a funcionalidade '**plano de manutenção**' conta com **tarefas predefinidas**, na rotina de um administrador de banco de dados, que envolve a manutenção de ambientes de dados.

Essa funcionalidade é uma forma prática de otimizar as tarefas funcionais de caráter cotidiano. Para saber mais sobre o assunto, indicamos que assista ao vídeo [dicas para um DBA Iniciante - Criando um plano de manutenção](#). Vale a pena conferir!

Até aqui você pôde entender sobre os processos mais importantes para a manutenção de banco de dados, sobretudo, em relação às tarefas preliminares e os comandos gerenciais do SQL Server.

Agora você poderá explorar, sob uma ótica mais prática, as funcionalidades mais comuns durante o processo de manutenção de banco de dados.

Acesse o link e saiba mais sobre a [aplicação do SPACEUSED](#).

Anexar, desanexar, importação, exportação e backup de dados - um exercício

Você viu que existem os processos principais de manutenção de banco de dados. Em função disso, para que você tenha mais prática desses procedimentos, vamos verificar um pouco mais sobre estas funcionalidades! Confira-as na sequência.

Anexar um banco de dados

Para um maior entendimento dessa técnica, é interessante que você **baixe** os seguintes arquivos para realizar o exercício. Lembramos de que eles precisam ser **salvos** em uma pasta chamada **LOCADORA**, em sua unidade C.

 **LOCADORA_DADOS.mdf**

 **LOCADORA_LOG.ldf**

Outro grupo de arquivos que também devem ser **baixados** são as planilhas de Excel, que serão importadas. Lembramos que as planilhas devem ser **salvas** em uma pasta chamada **IMPORTACAO**, em sua unidade C.

 **Tbl_Clientes**

 **Tbl_DetalhesDoPedido**

 **Tbl_Filmes**

 **Tbl_Genero**

 **Tbl_Pedidos.**

Quando procedemos uma operação de anexar um banco de dados, os arquivos de dados (arquivos MDF e LDF) devem estar disponíveis, preferencialmente, em **uma mesma pasta**. Do contrário, será preciso especificar o caminho atual do arquivo, o que resultará em mais trabalho.

Agora, veja neste link [como anexar um banco de dados usando o T-SQL](#).

Desanexar um banco de dados

Uma vez realizado o processo de anexar um banco de dados, é importante que você também saiba como desanexá-lo.

Ao desanexar, não se esqueça que a instância do SQL Server será **removida**, contudo, ela preserva o banco de dados e os arquivos.

Observe a **sintaxe** dessa ação:

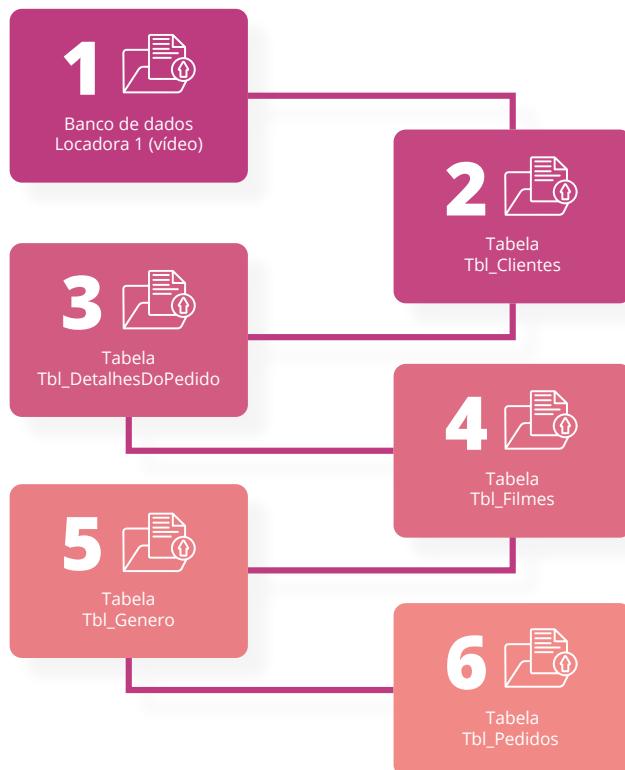
```
SP _ DETACH _ DB @dbname = 'nome _ banco _ de _ dados',
@skipchecks = 'verdadeiro/falso (true/false)',
@keepfulltextindexfile = 'verdadeiro/falso (true/false)',
```

Agora, acompanhe a demonstração no material sobre como **desanexar um banco de dados via T-SQL**.

Criando BD e tabelas

Agora chegou o momento de você criar um banco de dados! Preparado?

Vamos chamá-lo de Locadora1. Nesse BD, serão desenvolvidas **cinco tabelas**, cujos dados serão importados posteriormente. Acompanhe e repita a sequência abaixo, clicando em cada uma das etapas:



Agora, após realizada a criação do BD e das tabelas, precisamos importar os dados via SSMS. No próximo tópico, veremos como realizar esse procedimento, aprofundando ainda mais seus conhecimentos sobre como administrar corretamente seu banco de dados. Vamos lá!

Importar dados via SSMS

O SQL Server pode criar pacotes simples que importem ou exportem dados, entre seus vários formatos, como os bancos de dados, planilhas e arquivos de texto.

Confira com atenção a **sequência** que o ajudará a entender esse processo de forma mais simples, clicando em cada uma das etapas:



Agora que já importamos os dados, vamos analisar sobre o ponto de vista de exportar os dados. Acompanhe!

Exportar dados

Agora você está conhecendo um pouco mais sobre a exportação de dados, maneira pela qual é possível identificar quais são os **procedimentos de exportação** de dados via SSMS e T-SQL.

No próximo material acompanhe a [demonstração da exportação de dados via SSMS](#).

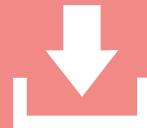
Para o **T-SQL**, usamos o utilitário **Bulk Copy Program (BCP)**. Clique e saiba mais sobre [exportação de dados via T-SQL](#).

Cabe ressaltar que o utilitário BCP não requer o conhecimento de T-SQL.

O BCP é um utilitário de cópia em massa que serve para duas finalidades. Vamos ver quais são:



Exportar dados em massa de uma tabela ou consulta SQL Server para um arquivo de dados.



Importar dados em massa de um arquivo de dados para uma tabela.

A sintaxe do BCP possui diversos argumentos. No entanto, como abordaremos a exportação, utilizaremos a **sintaxe** conforme segue.

```
bcp nome _ banco _ dados. schema.nome _ tabela/nome _ view/  
consulta  
out caminho _ do _ arquivo  
-U id _ login  
-C  
-T  
-U id _ login  
-C
```

Saiba mais



Para conhecer um pouco mais sobre esse processo observe no vídeo a seguir [BCP Utility - Export SQL Server data to CSV by BCP Utility](#) e descubra mais sobre sua utilização.

Agora que você aprendeu sobre como exportar e importar dados, vamos analisar como podemos nos proteger e guardar os dados por meio do uso do backup.

Backup

Uma **estratégia de backup** é vital na administração de um banco de dados, em que existem **riscos de dados e trabalho perdido**.



#PraCegoVer

Imagen de um notebook, ao fundo diversas imagens holográficas surgindo na frente do notebook.

Desse modo, os sistemas de cópias de dados **garantem a manutenção de informações armazenadas**, preservando sua integridade. Para isso, é preciso investir em **manter uma cópia do banco de dados**, que possibilite a recuperação das informações, caso ocorra algum problema.

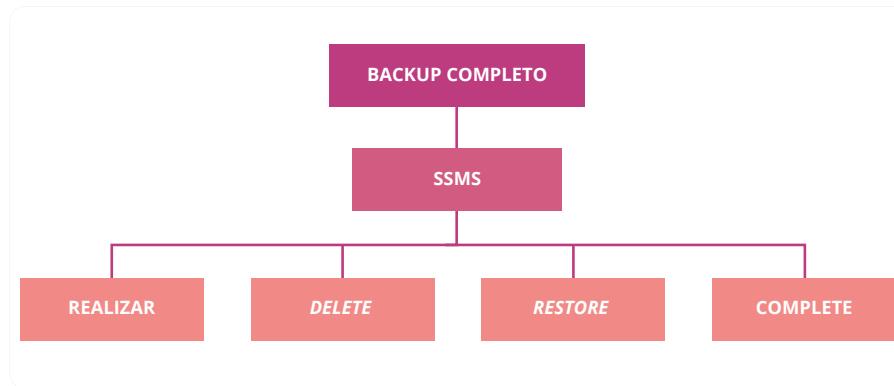


#PraCegoVer

Na imagem, existem dois ícones que representam dois bancos de dados. No da esquerda, há um título abaixo do ícone escrito "Criação do Banco de Dados" e no da direita, o título "Backup Completo". Em cada um, existem caixas de textos em que estão escritos "arquivo 1", "arquivo 2", "arquivo 3" e "arquivo 4". No meio deles, há uma seta da esquerda para a direita, que em cima dela, está escrito backup.

No modo de **backup completo**, os dados de um banco são definidos nos arquivos ou em grupos de arquivos. O backup completo possui *log* suficiente para **recuperar todos os dados**.

Observe um **esquema** de **backup completo** com suas definições.



#PraCegoVer

Imagen mostra o esquema do backup completo, passa pelo SSMS, e pelos itens realiza, delete, restore e consulta.

Agora observe os parâmetros estudados, na prática, por meio dos seguintes materiais:

[Backup Completo - SSMS - Realizar backup](#)

[Backup Completo - SSMS - Delete](#)

[Backup Completo - SSMS - Restore](#)

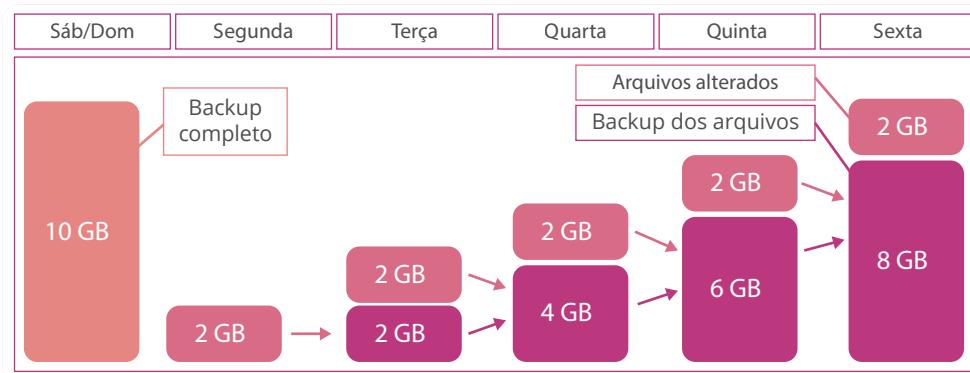
[Backup Completo - SSMS - Consulta](#)

Agora que conheceu um pouco mais sobre os formatos de backup que podem ser realizados, siga em frente para ver mais uma forma de backup que você pode utilizar em sua jornada profissional como um administrador de banco de dados.

Backup diferencial

O backup diferencial é realizado após o último backup completo, criando cópias apenas de **dados alterados**.

Observe a imagem abaixo com atenção.



#Pracegover

Na imagem, há um fluxograma representando o backup diferencial, o qual realiza o backup em dias da semana. Acima dele, estão escritos, da esquerda para a direita, os dias da semana começando por Sábado/Domingo, seguido de Segunda, Terça, Quarta, Quinta e Sexta. Embaixo, existem seis colunas com os dizeres backup completo (10 GB) abaixo de Sábado/Domingo; arquivos alterados (2 GB como parte de cada coluna, de Segunda a Sexta); e backup dos arquivos começando de Terça a Sexta com, respectivamente, 2 GB, 4 GB, 6 GB e 8 GB.

Aqui você notou que o backup completo do sistema acontece no sábado e domingo, quando os dados são integralmente copiados.

A partir desse momento, na segunda-feira, somente os dados que sofreram alterações são copiados, e assim sucessivamente.

De modo em geral, os dados alterados no dia, acrescidos dos dados do dia anterior, serão copiados, até que, ao final do ciclo semanal, um novo backup completo ocorra.

Acompanhe, com mais detalhes, um processo de backup diferencial nos links indicados:

[Backup Diferencial - SSMS - Criar tabela](#)

[Backup Diferencial - SSMS - Realizar](#)

Você estudou sobre o backup diferencial, aplicado depois do backup completo. Além disso, é necessário que você também veja sobre outro aspecto importante da administração do banco de dados: a ferramenta de segurança Docker. Acompanhe com atenção no próximo tópico.

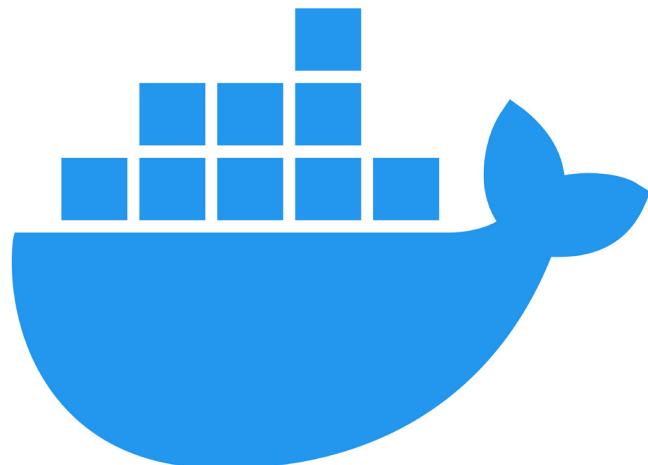
A ferramenta de segurança Docker



#Pracegover

Na imagem, há uma porta sendo enquadrada e, no meio dela, há a logo da empresa docker, que é uma baleia com vários *containers* em cima dela.

Os desenvolvedores e DBAs possuem **preocupações com a segurança** de seus backups. Por isso, a cada dia, são criadas ferramentas auxiliares, como o **Docker**, um pacote de aplicações que vêm em formas de *containers*, de bibliotecas isoladas e portáveis, adaptadas a cada necessidade. Ele é uma **plataforma open source** na **linguagem de programação Go**, que possui alto desempenho e foi **desenvolvida pela Google**.



#Pracegover

A logo da empresa Docker, que é uma baleia com vários *containers* em cima dela.

Sendo assim, ele **aborda diversas partes de softwares** de um sistema de arquivos completos com os recursos necessários para sua operação. Por isso, é uma plataforma de *containers*.



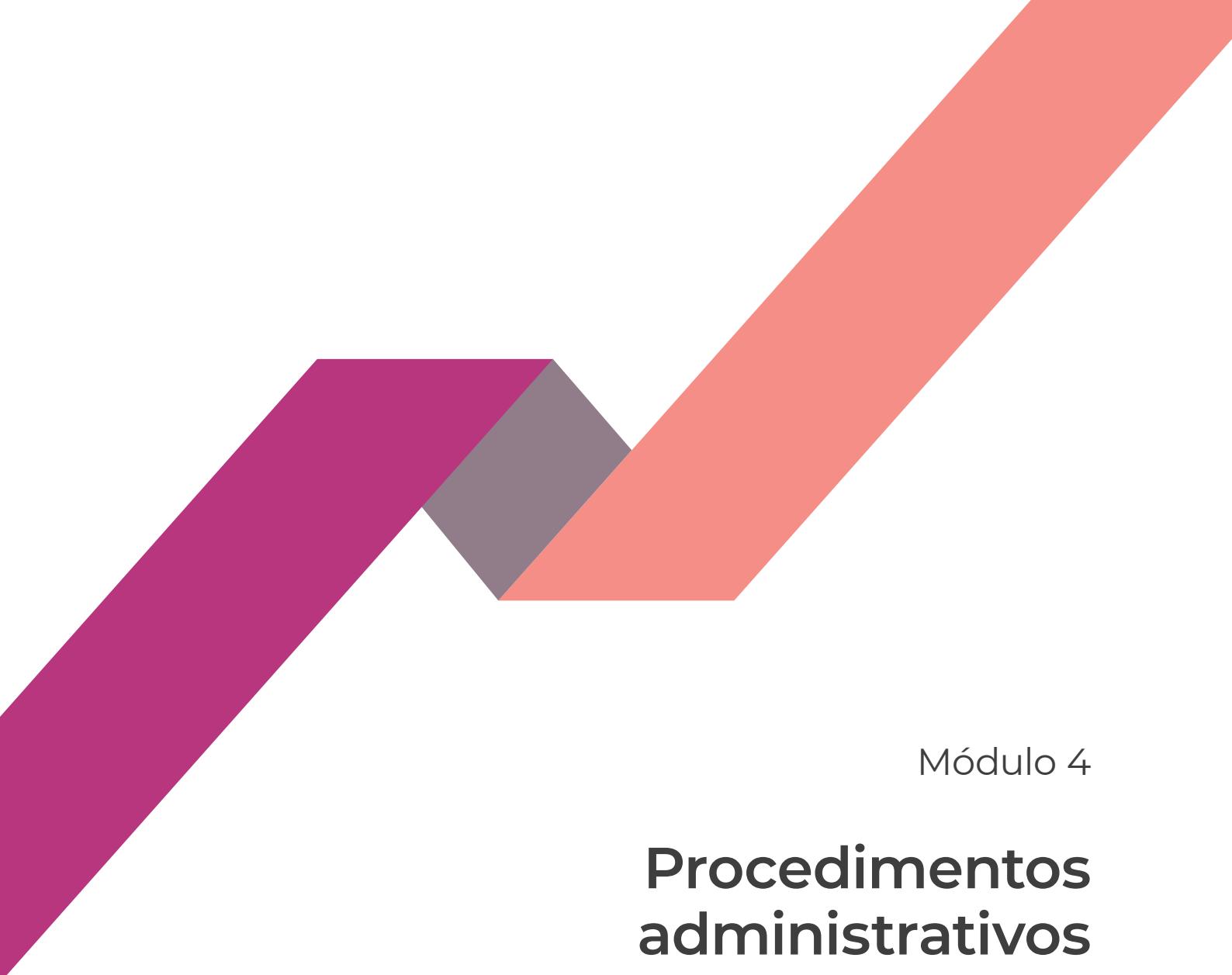
Saiba mais

Interessante, não é mesmo? Caso queria saber mais como funciona esse backup, indicamos o vídeo [backup e restore do SQL Server executando no Docker Container](#). Vale a pena conferir!

Parabéns! Você finalizou o Módulo 3!

Neste módulo, você aprimorou seus conhecimentos sobre dois processos relevantes, o gerenciamento e a manutenção de banco de dados. Para isso, você viu as tarefas preliminares e os comandos gerenciais do *SQL Server*, principalmente, envolvendo as ações de anexar, desanexar, importar e exportar um backup de dados.

No próximo módulo, você entenderá sobre os procedimentos administrativos associais ao banco de dados. Vamos lá?



Módulo 4

Procedimentos administrativos

Procedimentos administrativos

Neste último módulo, apresentaremos os **procedimentos administrativos** do banco de dados. Ou seja, os processos ligados à criação de um *logon* para um usuário conectar e utilizar um banco de dados, que são muito relevantes, uma vez que esses processos impactam diretamente a funcionalidade dos bancos de dados.

Logons e usuários



#Pracegover

Na imagem, há um notebook aberto e sendo enquadrado com duas mãos interagindo nele. Ao lado, há um smartphone em um suporte. De modo sobreposto, há uma projeção de uma janela com um cadeado e os dizeres "username", "apóstrofes" e o botão "OK".

A **criação de usuários e senhas** é fundamental em um banco de dados, para que somente quem tenha **permissão** consiga utilizá-lo. Portanto, é importante que você saiba como criá-los. Acompanhe com atenção!

usuário

O **usuário** de banco de dados é uma **identificação do logon**. Eles são criados nesse **nível** de banco de dados.

LOGONS

Os **logons** são criados no **nível da instância do banco de dados do servidor**. Eles permitem que **usuários se conectem aos serviços** ou instâncias do SQLServer. E ainda, desde que os usuários estejam **mapeados** para um *logon*, eles têm a permissão de se conectar à instância e utilizar os recursos do banco de dados. Além disso, é possível ter **logon sem definição**, criado sem o acesso a um banco de dados.

Na sequência abaixo, é possível acessar os links que demonstram como criar *logons* e usuários. Confira com atenção clicando nas caixas.

*Logons***Criar**

Podemos criar um *logon* com base em uma entidade de segurança do Windows. Como um usuário de domínio ou um grupo de domínio do Windows, ou com base no SQL Server.

**Verificar**

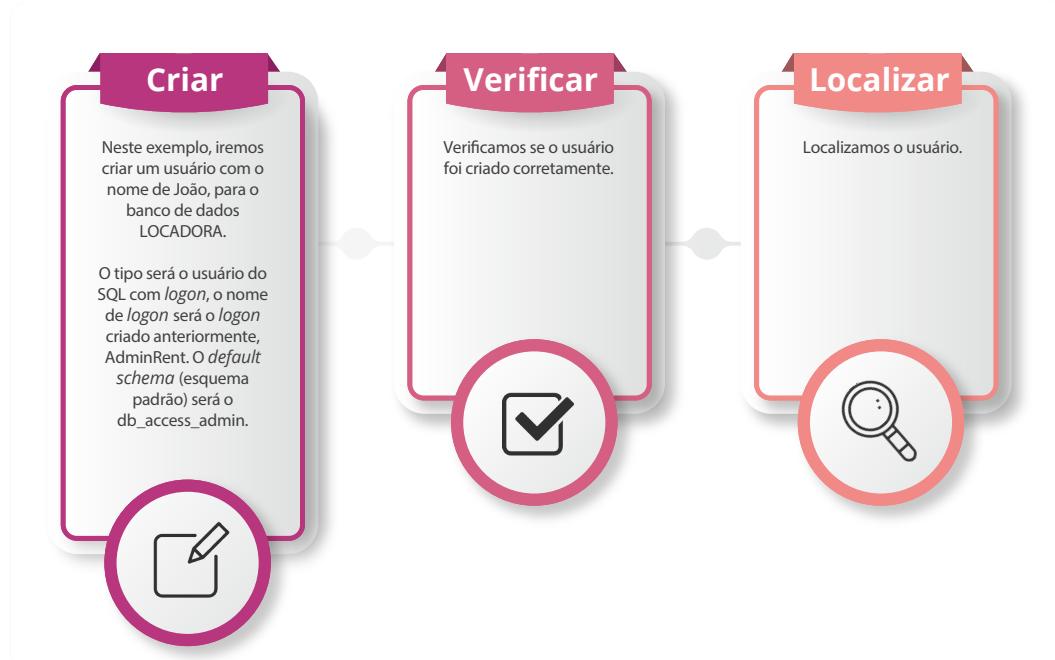
Verificamos se o *logon* foi criado corretamente.

**Localizar**

Localizamos o *logon* por meio do T-SQL.



Usuários



Viu só a importância das instâncias e de seus *logons*? Agora você começará a ver um pouco mais sobre os arquivos de *log* que podem ser gerados. Vamos lá?

Arquivos de *log*

Confira abaixo as ações necessárias para usar o recurso dos *logs* clicando nas caixas.



Agora que você viu como gerar os arquivos de *logs*, a seguir você analisará mais a fundo sobre o monitoramento de desempenho do servidor.

Monitoramento de desempenho e de atividade de servidor



#Pracegover

Na imagem, há um notebook aberto sendo enquadrado com duas mãos interagindo nele. De modo sobreposto, existem vários gráficos sendo projetados de cores claras.

O monitoramento do desempenho e da atividade do servidor são tarefas de **muita importância** no cotidiano do DBA. Assim, um monitoramento efetivo destaca a **criação de relatórios** periódicos e instantâneos do desempenho atual, possibilitando o isolamento de processos que possam gerar problemas. Um bom monitoramento realiza uma **coleta constante de dados**, que auxilia no controle das tendências de desempenho.

É importante que você saiba que, além do *SQL Server*, o Windows oferece outros utilitários, que permitem monitorar e exibir a condição atual do banco de dados e controlar o desempenho, conforme as condições se alteram.

Convidamos você a acessar o material **monitorar o banco de dados**, para entender o desempenho e a eficácia dele a partir do programa *SQL Server*.

Saiba mais



Uma plataforma interessante para o **monitoramento de desempenho do servidor** em forma visual é o **Enterprise Manager**. Muito visual e com possibilidade de monitoria via web, otimiza tempo e diminui as tarefas manuais dessa atividade. Ele utiliza um **servidor de aplicação e o banco de dados repositório**, onde as informações de monitoria dos bancos de dados acompanhados serão armazenadas.

Existem diversas outras ferramentas que podem ser úteis. Para saber mais, indicamos os vídeos sobre [ferramentas de monitoração para o SQL Server - Parte 1](#) e [ferramentas de monitoração do SQL Server - Parte 2](#).

Neste tópico você estudou acerca de como monitorar o desempenho e a atividade de um servidor ligado a um banco de dados. Nos recursos apresentados, foi possível entender quais são as etapas voltadas para a realização desse monitoramento.

Agora, chegou o momento de abordar outro assunto importante: a gestão de segurança.

Gestão da segurança

Neste tópico, você estudará os conhecimentos sobre as ações e as medidas ligadas à gestão de segurança, principalmente, no contexto do SQL.



#Pracegover

Na imagem, as mãos de um programador estão sendo enquadradas enquanto digitam em um teclado. Na frente dele, há uma tela de computador mostrando um ícone de um escudo com uma fechadura no meio. Atrás da tela, há um fundo escuro com linhas de código de programação.

O **framework de segurança** do SQL atua junto a uma **engine**, que gerencia a segurança de acordo com uma **hierarquia de permissões** aplicadas nos objetos do banco de dados (entidades), chamados de **Securables** (tabelas, servidor, banco de dados, views etc). **Securables** no SQL possuem três escopos de permissão, que seriam o servidor, banco de dados e *schema*; no nível do Windows incluem, ainda, arquivos e chaves de registro.

Os **operadores** se referem aos usuários, grupos e aplicações, como *Principals*, que são entidades autenticadas para atuar no nível do sistema operacional, do SQL e do banco de dados. Por exemplo, no Windows, o principal pode ser um usuário, no SQL um *login*, e no banco de dados um usuário ou aplicação.

As permissões trabalham com os acessos aos *principals* e *securables*, tornando possível a remoção ou aplicação de permissões no nível do servidor e banco de dados. O nível do servidor se aplica a *logins*, credenciais e *roles* de servidor. Em nível banco de dados, se aplicam aos usuários, *schemas* e *roles* de banco de dados.

Com o entendimento desses elementos associados à gestão de segurança, sobretudo, em relação aos conceitos de *frameworks* de segurança e os operadores, podemos nos aprofundar um pouco mais, principalmente nos conceitos direcionados à gestão de segurança no nível do banco de dados. Acompanhe!

Gestão de segurança no nível do banco de dados

Após essa breve introdução acerca dos conceitos principais da gestão de segurança no contexto SQL, neste tópico, vamos nos debruçar na gestão de segurança voltada especificamente aos bancos de dados.

Os métodos de autenticação do SQL podem utilizar logins do Windows ou do próprio SQL. É possível configurar o acesso ao SQL, a partir da autenticação do Windows, com o uso do *Active Directory* e seu controle de senhas ou, ainda, utilizar a autenticação mista, tanto do Windows quanto do SQL.

A gestão de *logins* permite que os acessos possam ser criados a partir do Windows ou do SQL.

Chaves, certificados e criptografia

As chaves são valores que, aplicados às **funções de criptografia e de descriptografia** de uma informação, podem ser caracterizadas como uma **chave simétrica ou assimétrica**.

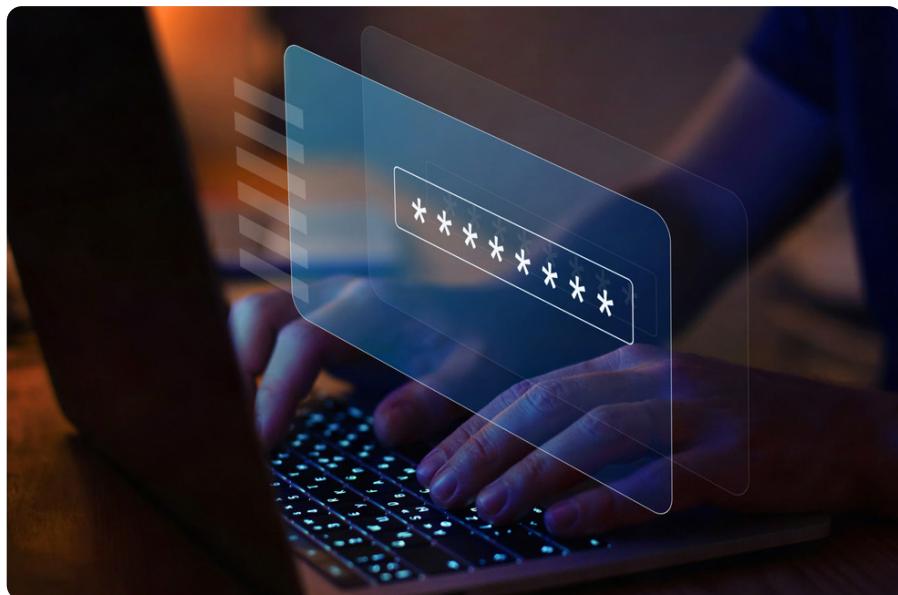
Chave simétrica

É utilizada para criptografia e descriptografia.

Chave assimétrica

Mais usada em DBA, é composta por um par de valores: a chave privada (criptografada) e a chave pública (descriptografada).

Já o **certificado** associa uma **chave pública com a entidade gerenciadora** de seu conteúdo, como o tema, a identificação da entidade, a validade, a assinatura digital, entre outros.



#Pracegover

Na imagem, as mãos de um programador estão sendo enquadradas enquanto digita no teclado de um notebook. Entre eles, há uma sobreposição de uma imagem de uma caixa de texto em que há seis asteriscos escritos.

A criptografia de informações no SQL Server, por sua vez, ocorre por uma **hierarquia de criptografias**, validadas por uma estrutura de gestão de chaves. Cada camada criptografa a camada anterior, por meio de combinações de certificados, chaves assimétricas e chaves simétricas.

O recurso **TDE (Transparent Data Encryption)** no SQL realiza a criptografia e descriptografia de dados e log em **tempo real** com mais **segurança**.

Exemplo



Imaginemos que um HD foi roubado; caso esteja protegido, não poderá ser lido e utilizado. Em casos assim pode-se perceber a importância do TDE. Para validação dele, alguns passos devem ser seguidos. Observe:



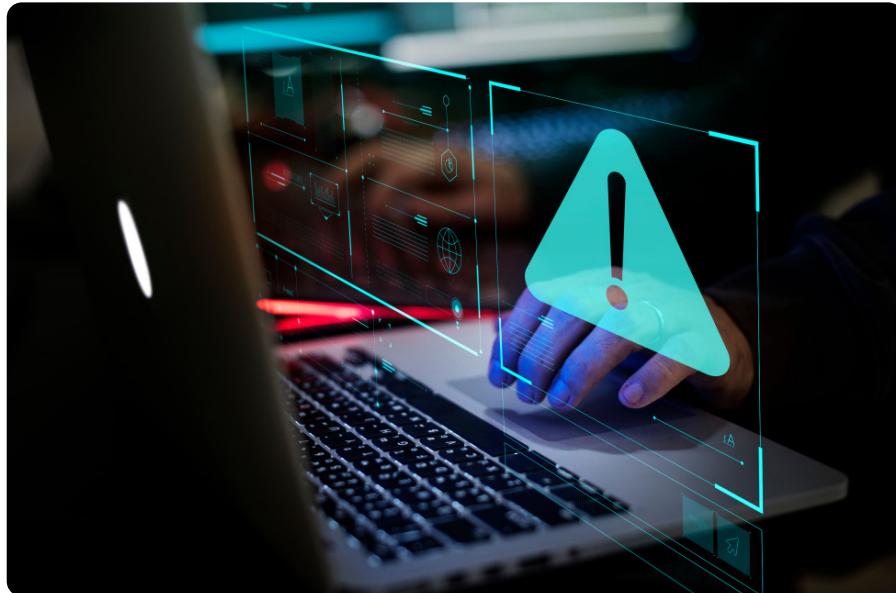
Vale lembrar que esse recurso está disponível apenas no *SQL Server Developer*.

Neste tópico, você aprendeu sobre os conceitos de chaves e certificados, os quais são fundamentais no processo de gestão de segurança de banco de dados. Somado a isso, vimos o conceito de criptografia, bem como observamos como ele é aplicado e validado.

No próximo tópico, vamos compreender melhor sobre um processo muito relevante para a gestão de segurança de banco de dados: a auditoria de segurança.

Auditoria de segurança

O **SQL Server** oferece as **ferramentas** para auditoria das atividades no banco de dados, permitindo que o DBA elabore estratégias de segurança.



#Pracegover

Imagen de uma pessoa digitando em um notebook, um holograma aparece na frente do notebook com a imagem de um ponto de exclamação.

Uma boa maneira de monitorar e criar essas estratégias é observar o **comportamento do usuário no sistema**. Um rastreamento de auditoria permite levantar o que foi feito, e a localização (máquina) da ação realizada.

O SQL Server *Profile* realiza **traces**, que são **conjuntos de instruções capturadas em tempo real** reunidas em *templates* reutilizáveis que monitoram as atividades. Assim, o *trace* armazena os dados e os confere periodicamente.

Uma **auditoria** pode ocorrer por meio de DDL *Triggers*, que **disparam quando certas alterações são executadas** e, segundo os parâmetros estabelecidos, **impedem que as alterações estruturais sejam feitas**, registrando, inclusive, o usuário que tentou implementar as alterações.

Existe também o **SQL Server Audit** que realiza as **buscas** e pode logar eventos na instância, monitorando em nível servidor e banco de dados. Disponível no *SQL Server Developer*.

Observou todos os detalhes que compõem essa etapa de administrar um banco de dados? Com isso, é possível monitorar as mais diversas situações dentro deste campo.

Fechamento

Parabéns! Você finalizou o curso Administrando Banco de Dados!

Aqui, você estudou sobre o contexto de atuação do administrador de banco de dados, assim como todos os conceitos atrelados à estruturação física e lógica de um banco de dados.

Em seguida, abordamos os processos de gerenciamento e de manutenção do sistema de banco de dados e ações posteriores à criação deles. Já no último módulo, você viu os procedimentos administrativos, sobretudo, ligados à interação com os usuários desses bancos, além de ações voltadas à segurança do banco de dados. Foi possível, também, aprofundar seu aprendizado quanto à **segurança e proteção** que os dados devem ter dentro das organizações, além de conhecer seus principais conceitos.

Lembre-se! O importante é **sempre praticar e se manter atualizado** a respeito dos assuntos aqui apresentados. Isso porque os avanços tecnológicos e as alterações na área são constantes em nosso dia a dia!

Esperamos que seus estudos não acabem por aqui! Continue pesquisando sobre o assunto, pois essa área requer a contínua busca por aperfeiçoamento e atualização.

Até a próxima!

Referências

BOGHI, Cláudio. **Ranking de Sistemas de Bancos de Dados mais usados em 2019/2020.** São Paulo, 2020. Disponível em: <https://www.redetv.uol.com.br/columnistas/desvendando-a-tecnologia/ranking-de-sistemas-de-bancos-de-dados-mais-usados-em-2019-2020>. Acesso em: 27 jan. 2021.

DEVMEDIA. **Guia Completo de SQL. Devmedia.** São Paulo, 2018. Disponível em: <https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>. Acesso em: 26 jan. 2021.

DEVMEDIA. **Introdução ao Cassandra.** São Paulo, 2017. Disponível em: <https://www.devmedia.com.br/introducao-ao-cassandra/38377>. Acesso em: 23 jan. 2021.

FERNADES, Henrique Marques. **Banco de Dados Relacional (SQL) e Não Relacional (NoSQL) – O que são, para que servem e qual a diferença?** Henrique Marques Fernandes. 2020. Disponível em: <https://marquesfernandes.com/tecnologia/banco-de-dados-relacional-sql-e-nao-relacional-nosql-o-que-sao-para-que-servem-e-qual-a-diferenca/>. Acesso em: 28 jan. 2021.

FREELANCERMAP. **O que faz um Administrador de Bancos de Dados? freelancermap.** São Paulo, 2020. Disponível em: <https://www.freelancermap.com/blog/pt/que-faz-administrador-bancos-dados/>. Acesso em: 26 jan. 2021.

FREEPIK. **Recursos gráficos para todos.** Freepik, 2021. Disponível em: <https://br.freepik.com/>. Acesso em: 28 de out. de 2021.

GOMES, Pedro César Tebaldi. **Quais os principais bancos de dados e quais suas diferenças?** Porto Alegre, 2019. Disponível em: <https://www.opservices.com.br/banco-de-dados/>. Acesso em: 23 jan. 2021.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais.** São Paulo: Pearson Prentice Hall, 2010.

LUCIDCHART. **O que é um esquema de banco de dados?** LUCIDCHART, 2019. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-um-esquema-de-banco-de-dados>. Acesso em: 26 jan. 2021.

METTZER. **O melhor editor para trabalhos acadêmicos já feito no mundo.** Mettzer. Florianópolis, 2016. Disponível em: <http://www.mettzer.com/>. Acesso em: 21 ago. 2016.

ORACLE. **Modelos de Implementação de Banco de Dados em Nuvem.** Oracle. São Paulo, 2018. Disponível em: <https://www.oracle.com/br/database/what-is-a-cloud-database/>. Acesso em: 27 jan. 2021.

PASCUTTI, Márcia Cristina Dadalto. **Administrador de banco de dados. Paraná: IFPR, 2012.**

SHUTTERSTOCK. **Transforme ideias em conquistas.** ShutterStock, 2021. Disponível em: <https://www.shutterstock.com/pt/>. Acesso em: 28 de out. de 2021.

