

Aula 01

*TJM-SP (Analista em Comunicação e
Processamento de Dados) Banco de
Dados*

Autor:
André Castro

23 de Julho de 2024

Índice

| | |
|---|-----|
| 1) Microsoft SQL Server 2012 ou Superior | 3 |
| 2) Conceitos sobre o Database Engine | 16 |
| 3) Construindo um Banco de Dados | 36 |
| 4) SQL Server na Nuvem | 63 |
| 5) Questões Comentadas - Microsoft SQL Server - CEBRASPE | 90 |
| 6) Questões Comentadas - Microsoft SQL Server - Multibancas | 105 |
| 7) Lista de Questões - Microsoft SQL Server - CEBRASPE | 147 |
| 8) Lista de Questões - Microsoft SQL Server - Multibancas | 155 |



MICROSOFT SQL SERVER 2012 OU SUPERIOR

Esta aula tem por objetivo fazer uma rápida introdução aos conceitos e às características do SQL Server que aparecem em provas de concurso. Semelhante à maioria dos sistemas gerenciadores de banco de dados relacionais (SGBDs), o SQL Server inclui vários componentes utilizados para o gerenciamento da base de dados. O produto em si, no entanto, é frequentemente dividido em duas categorias distintas: Business Intelligence (BI) e Database Engine (ou motor de banco de dados).

Os recursos de BI do SQL Server 2012 podem agregar valor, sendo altamente visíveis e eficazes para usuários empresariais e consumidores de dados. Nesta aula, nós vamos nos concentrar nas características específicas do Database Engine.

Ao longo da aula, faremos várias questões para fixar o conhecimento. Vale ressaltar que várias questões abordam o SQL 2008, mas o SGBD evoluiu sem perder funcionalidades antigas.

HISTÓRICO DAS VERSÕES DO SQL SERVER

Neste primeiro capítulo da aula, trazemos uma breve evolução das versões do SQL Server a partir do SQL Server 2012. Assim, você terá condições de entender como houve a evolução deste SGBD. Não há necessidade de memorizar todos os termos, mas sua leitura facilitará o entendimento dos recursos disponíveis no SQL Server.

Veja na tabela abaixo as melhorias trazidas por cada versão do SQL Server:

| Versão | Desempenho e escalabilidade | Suporte e diagnóstico |
|-----------------|--|---|
| SQL Server 2012 | Procedimento de limpeza do agente de distribuição aprimorado; Escala Dinâmica de Objetos de Memória; Habilitação > 8 TB para Buffer Pool (espaço de endereço virtual de 128 TB); limpeza de controle de alterações do Change Tracking. | Suporte de dumps completos para Agentes de Replicação; Diagnóstico aprimorado no XML do showplan; Melhor correlação entre diagnósticos XE e DMVs; Melhor diagnóstico de concessão / uso de memória; Adiciona rastreamento de protocolo a etapas de negociação SSL; Configuração do nível de compatibilidade correto para o banco de dados de distribuição; Novo comando DBCC para clonar um banco de dados; Arquivo TempDB e informações sobre tamanho de arquivo no log de erros do SQL; IFI suporta mensagens no log de erros do SQL Server; Nova DMF para substituir o DBCC INPUTBUFFER (Função de |



| | | |
|------------------------|---|--|
| | | Gerenciamento Dinâmico); Aprimoramento de XEvents para falha de roteamento de leitura para um grupo de disponibilidade; Manipulação aprimorada do Service Broker com o failover do grupo de disponibilidade; Particionamento Automático do Soft-NUMA. |
| SQL Server 2014 | Particionamento Automático Soft NUMA; Extensão do pool de buffers; Escala de objeto de memória dinâmica; Sugestão MAXDOP para comandos DBCC CHECK; Melhoria do spinlock SOS_RWLock; Implementação Espacial Nativa (desempenho de consultas espaciais). | Logout de tempo limite de AlwaysON; AlwaysON XEvents e contadores de desempenho; Altera a limpeza de rastreamento; Clonagem de banco de dados; Adições DMF; DMF para recuperar o buffer de entrada no SQL Server; Suporte DROP DDL para Replicação; Privilégio IFI para conta de serviço SQL; Concessões de memória - problemas de manipulação; Execução de consulta leve por perfil de operador; Diagnósticos de execução de consulta; Diagnóstico de execução de consulta para derramamento de tempdb; Suporte a tempdb |
| SQL Server 2016 | Procedimento de limpeza do DB de distribuição aprimorado; Altera o Tracking Cleanup; Usa o tempo limite da CPU para cancelar a solicitação do Administrador de Recursos; SELECT INTO para criar a tabela de destino no grupo de arquivos; Ponto de verificação indireto aprimorado para TempDB; Melhor desempenho de backup do banco de dados em grandes máquinas de memória; Suporte de compactação de backup de VDI para bancos de dados habilitados para TDE; Carregamento dinâmico de parâmetros do perfil do agente de replicação; Suporta opção | Suporte completo ao DTC para bancos de dados em um grupo de disponibilidade; Atualização para refletir com precisão o status de criptografia do TempDB; Novas opções para gerar clone e backup verificados; Suporte do Service Broker (SSB) para o DBCC CLONEDATABASE; Novo DMV para monitorar o uso do espaço de armazenamento da versão do TempDB; Suporte a Full Dumps para Agentes de Replicação; Aprimoramento de Eventos Estendidos para falha de roteamento de leitura para um Grupo de Disponibilidade; Novo DMV para monitorar o log de transações e informações de VLF; Informações do |



MAXDOP para criar / atualizar estatísticas; Atualização de Estatísticas Automáticas Melhorada para Estatísticas Incrementais

processador em sys.dm_os_sys_info; informações modificadas de extensão em sys.dm_db_file_space_usage; Informações de segmento em sys.dm_exec_query_stats; Configuração do nível de compatibilidade correto para o banco de dados de distribuição; Exposição das últimas informações válidas do DBCC CHECKDB; Aprimoramentos do showplan do XML; Suporte à Replicação para Bancos de Dados com Agrupamentos Suplementares de Caracteres; Manuseio adequado do Service Broker com o failover do grupo de disponibilidade; O paralelismo aprimorado aguarda a solução de problemas; Maior consistência entre os DMVs para as mesmas informações; Solução aprimorada de problemas de deadlocks de paralelismo intra-consulta; Recarga dinâmica de alguns parâmetros do perfil do agente de replicação

SQL Server 2017

Assemblies CLR podem ser adicionados a uma lista de desbloqueio; Recriação de índice online recuperável; Opção IDENTITY_CACHE para ALTER DATABASE SCOPED CONFIGURATION; nova geração de processamento de consultas; Ajuste automático de banco de dados; Novo gráfico de recursos de banco de dados; opção sp_configure para aprimorar a segurança de assemblies CLR; instalação agora permite especificar o tamanho inicial do arquivo tempdb de até 256 GB por arquivo; A coluna modified_extent_page_count rastreia mudanças diferenciais em cada arquivo de banco de dados; sintaxe SELECT INTO T-SQL agora suporta o carregamento de uma tabela em um grupo de arquivos; Suporte a transações cruzadas de banco de dados; Nova funcionalidade de grupos de disponibilidade; Novas visualizações de gerenciamento dinâmico; Opções adicionais para Database Tuning Advisor (DTA); Melhorias na memória; Novas funções de string; novas opções de acesso em massa; Aprimoramentos de objetos otimizados para memória; DATABASE SCOPED CREDENTIAL é uma nova classe de garantia; Banco de dados COMPATIBILITY_LEVEL 140 é adicionado.



SQL Server 2019
(ainda não lançada)

Suporte a UTF-8; Criação de índice online retomável; Construção e reconstrução de índice on-line em cluster columnstore; Sempre criptografado com enclaves seguros; Processamento Inteligente de Consultas; Extensão de programação de linguagem Java; Recursos do SQL Graph; Definição de configuração para operações DDL online e retomadas; Redirecionamento de conexão de réplica secundária; Descoberta e classificação de dados; Suporte expandido para dispositivos de memória persistente; Suporte para estatísticas columnstore no DBCC CLONEDATABASE; Novas opções adicionadas a sp_estimate_data_compression_savings; Clusters de failover dos Serviços de Aprendizado de Máquina; Infraestrutura de perfil de consulta leve habilitada por padrão; Novos conectores PolyBase; Nova função do sistema sys.dm_db_page_info retorna informações da página; Processamento de consulta inteligente adiciona inline escalonada UDF; Mensagem de erro de truncamento aprimorada; Suporte para agrupamentos UTF-8; Uso de tabela derivada ou exibição de aliases em consultas de correspondência de gráfico; Dados de diagnóstico aprimorados para o bloqueio de estatísticas; Buffer Pool Híbrido; Mascaramento de dados estáticos.

INSTALAÇÃO DO SQL SERVER

Neste capítulo da aula, vamos ver quais são os requisitos necessários de hardware e software para a instalação do SQL Server. Em seguida, vamos saber quais são as etapas para instalação do SQL Server 2017.

REQUISITOS DE HARDWARE E SOFTWARE PARA A INSTALAÇÃO DO SQL SERVER

Os seguintes requisitos se aplicam a todas as instalações:

| Componente | Requisito |
|-------------------------|--|
| .NET Framework | O RC1SQL Server 2016 (13.x) e posteriores exigem o .NET Framework 4.6 para o Mecanismo de Banco de Dados, Master Data Services ou Replicação. Windows 8.1, e Windows Server 2012 R2 exigem o KB2919355 antes de instalar o .NET Framework 4.6. |
| Software de rede | Os sistemas operacionais com suporte para SQL Server têm software de rede interno. |



| | |
|---------------------|---|
| Disco rígido | OSQL Server requer no mínimo 6 GB de espaço disponível no disco rígido. Os requisitos de espaço em disco variam de acordo com os componentes do SQL Server instalados. |
| Unidade | É necessária uma unidade de DVD, conforme apropriado, para a instalação a partir de disco. |
| Monitor | OSQL Server requer um monitor com resolução Super-VGA (800 x 600) ou superior. |
| Internet | A funcionalidade de Internet requer acesso à Internet |

Requisitos de processador, de memória e do sistema operacional

Os requisitos de memória e processador a seguir aplicam-se a todas as edições do SQL Server:

| Componente | Requisito |
|----------------------------------|---|
| Memória | Mínimo: Edições Express: 512 MB Todas as outras edições: 1 GB Recomendado: Edições Express: 1 GB Todas as outras edições: Pelo menos 4 GB e deve ser aumentado à medida que o banco de dados cresce para garantir um ótimo desempenho. |
| Velocidade do processador | Mínimo: processador x64: 1,4 GHz Recomendado: 2.0 GHz ou mais rápido |
| Tipo de processador | Processador x64: AMD Opteron, AMD Athlon 64, Intel Xeon com suporte Intel EM64T, Intel Pentium IV com suporte EM64T |

INSTALAÇÃO DO SQL SERVER 2017

Um dado importante, deve-se saber em qual ambiente deseja instalar o SQL Server. O mesmo possui algumas edições, cada uma com suas características e funcionalidades. Após atender aos requisitos, vamos partir para a instalação do SQL Server propriamente dito.



Center 1807 blog.

SQL Server Evaluations

| | |
|--|---|
|  SQL Server 2019 CTP Evaluations 180 days |  |
|  SQL Server 2017 RTM Evaluations |  |
|  SQL Server 2016 with SP2 Evaluations 180 days |  |
|  SQL Server 2014 SP3 Evaluations 180 days |  |

Neste momento, veremos a instalação do Mecanismo de Banco de Dados do SQL Server 2017. Veremos como instalar o SQL Server com o Assistente de Instalação, que é o mais recomendado. Aplica-se a SQL Server 2016 (13.x) e SQL Server 2017 (14.x).

O SQL Server Installation Center é o utilitário que você vai utilizar antes de instalar o SQL Server 2017. Ele ativa o SQL Server Setup para que você possa criar novas instâncias e outras funcionalidades.

A Evaluate Edition (versão gratuita) do SQL Server 2017 está disponível em:

<https://www.microsoft.com/pt-br/sql-server/sql-server-downloads>

Experimente o SQL Server na infraestrutura local ou na nuvem

SQL Server 2017 na infraestrutura local

Crie aplicações de missão crítica inteligentes usando uma plataforma de dados híbridos escalável para workloads exigentes. Comece a usar o trial gratuito de 180 dias do SQL Server 2017 no Windows.

Faça download do trial gratuito >

SQL Server na nuvem

Aproveite a alta disponibilidade, a segurança e a inteligência internas do Azure SQL Database e use o mecanismo familiar do SQL sem a complexidade do gerenciamento da infraestrutura. Comece a usar o SQL Database gratuito no Azure.

Comece gratuitamente >

Ou faça download de uma edição especializada gratuita

Developer

O SQL Server 2017 Developer é uma edição gratuita completa, licenciada para uso como banco de dados de desenvolvimento e teste em um ambiente de não produção.

Faça download agora mesmo ↓

Express

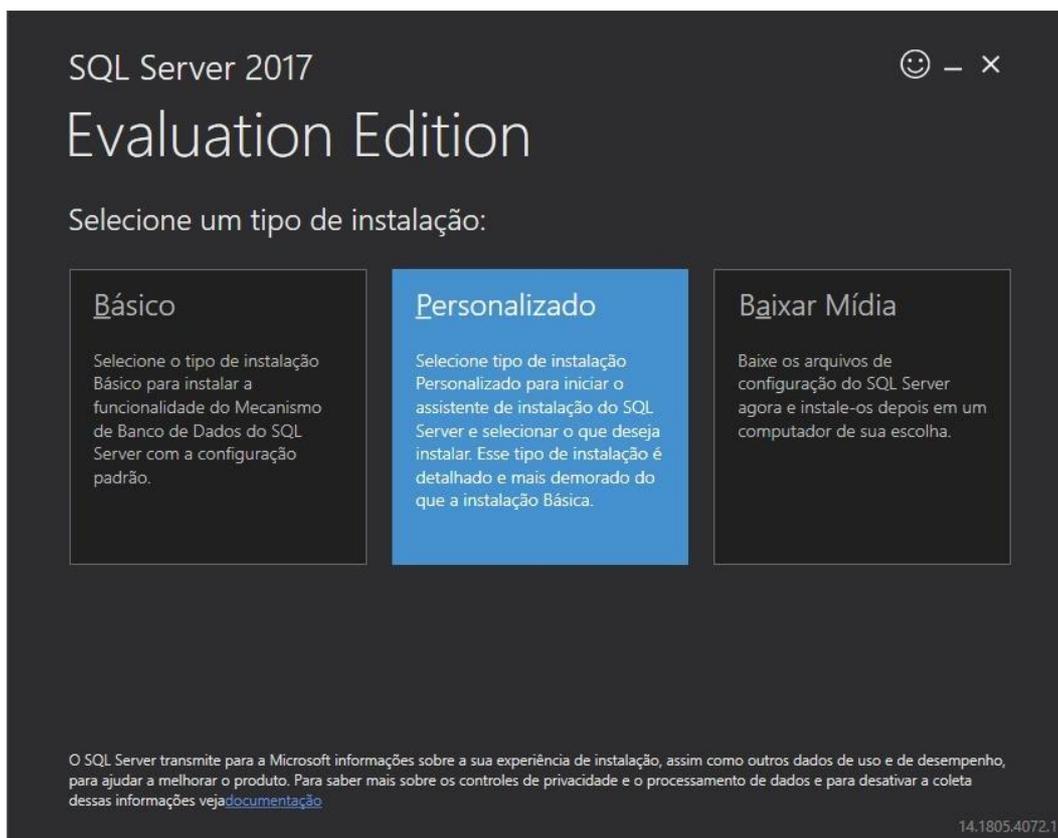
O SQL Server 2017 Express é uma edição gratuita do SQL Server, ideal para desenvolvimento e produção de aplicativos de área de trabalho, Web e pequenos servidores.

Faça download agora mesmo ↓

Selecione SQL Server 2017 na infraestrutura local. Em seguida, aparecerá uma tela para você fazer a opção por um tipo de instalação.

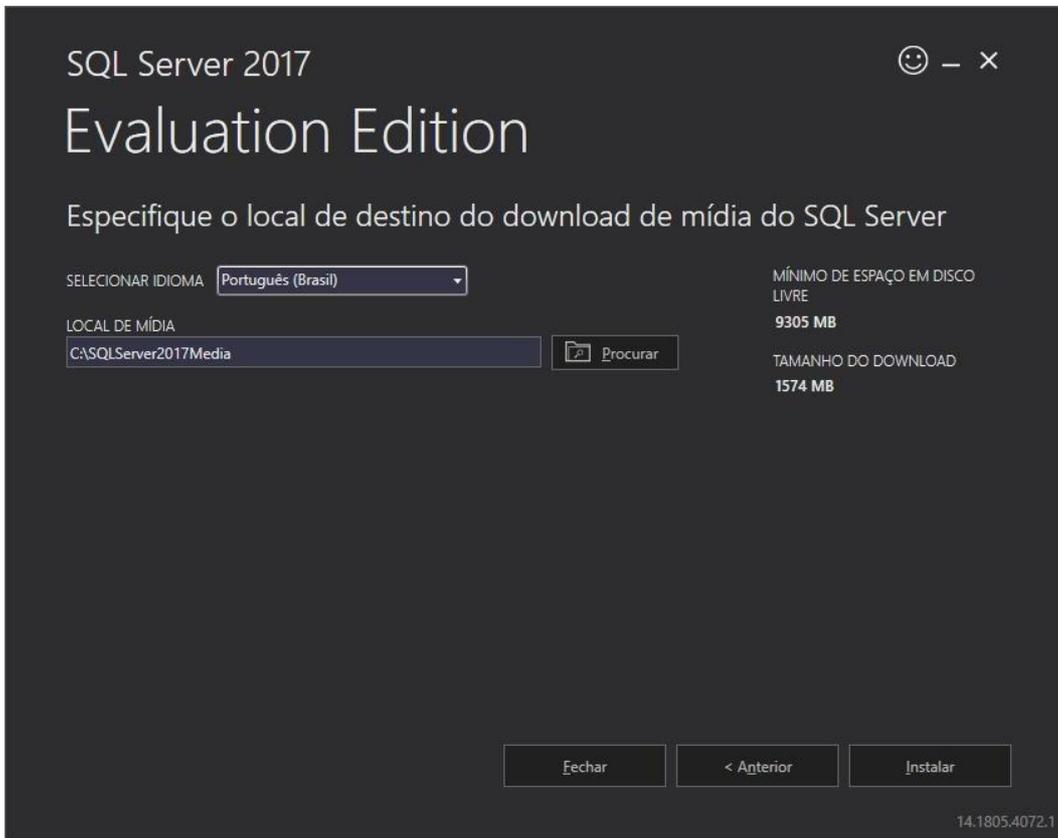


Para que você faça a instalação do SQL Server 2017 utilizando o assistente de instalação, selecione o tipo de instalação **Personalizado**:

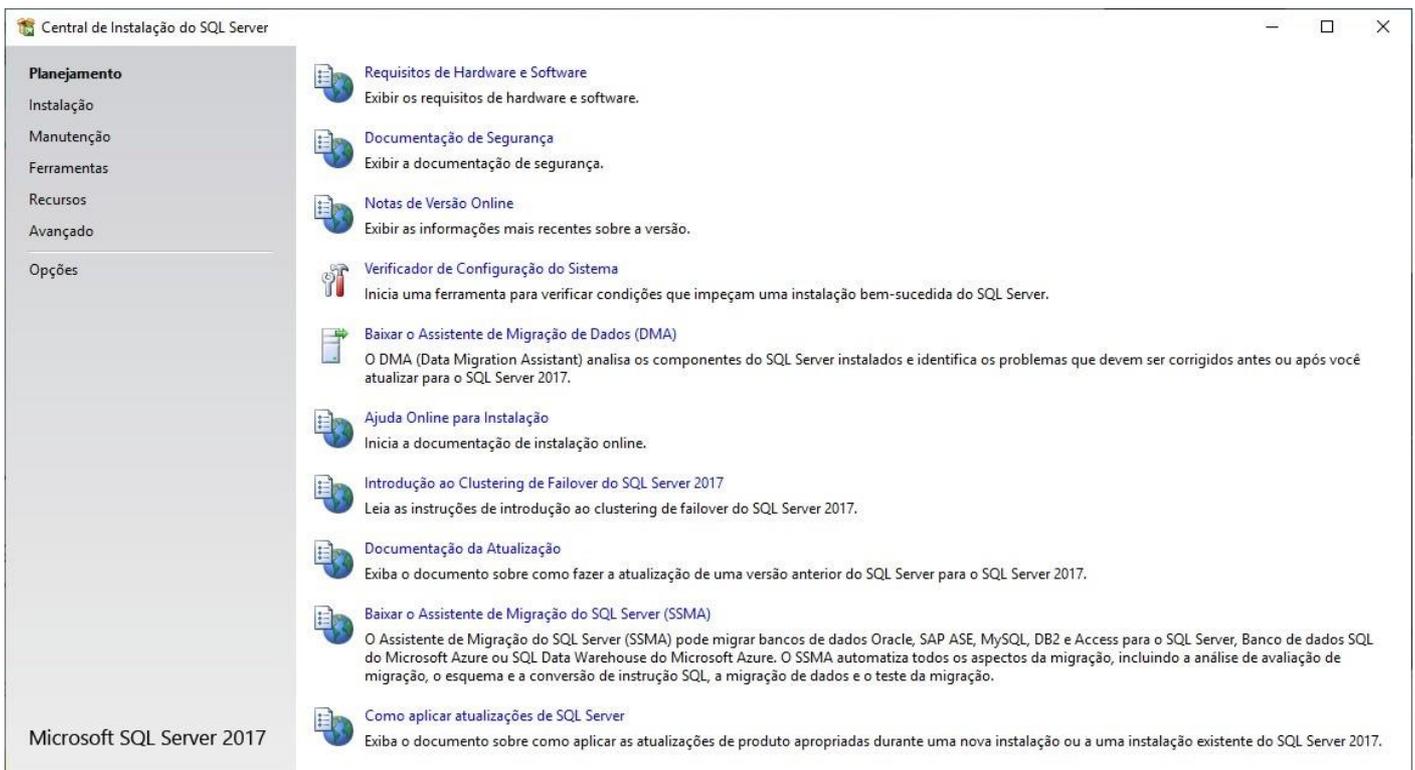


Na próxima etapa, você deverá especificar o local de destino do download:





A partir desse momento, aparece a tela com a Central de Instalação do SQL Server. Você deverá seguir as etapas indicadas para a instalação.



No menu disponível na Central de Instalação do SQL Server, é possível visualizar:

Planejamento: fornece links para planejar documentação e ferramentas de planejamento úteis

Instalação: Contém opções para instalar instâncias do SQL Server, atualizar a partir de versões anteriores e adicionar recursos

Manutenção: contém opções para atualizar sua edição de SQL Server, reparar sua instalação e remover nós de cluster

Ferramentas: Contém opções para verificar a configuração do sistema antes da instalação, relatar a configuração atual dos produtos SQL Server e atualizar pacotes de Integration Services

Recursos: Fornece links para documentação adicional que pode ser útil

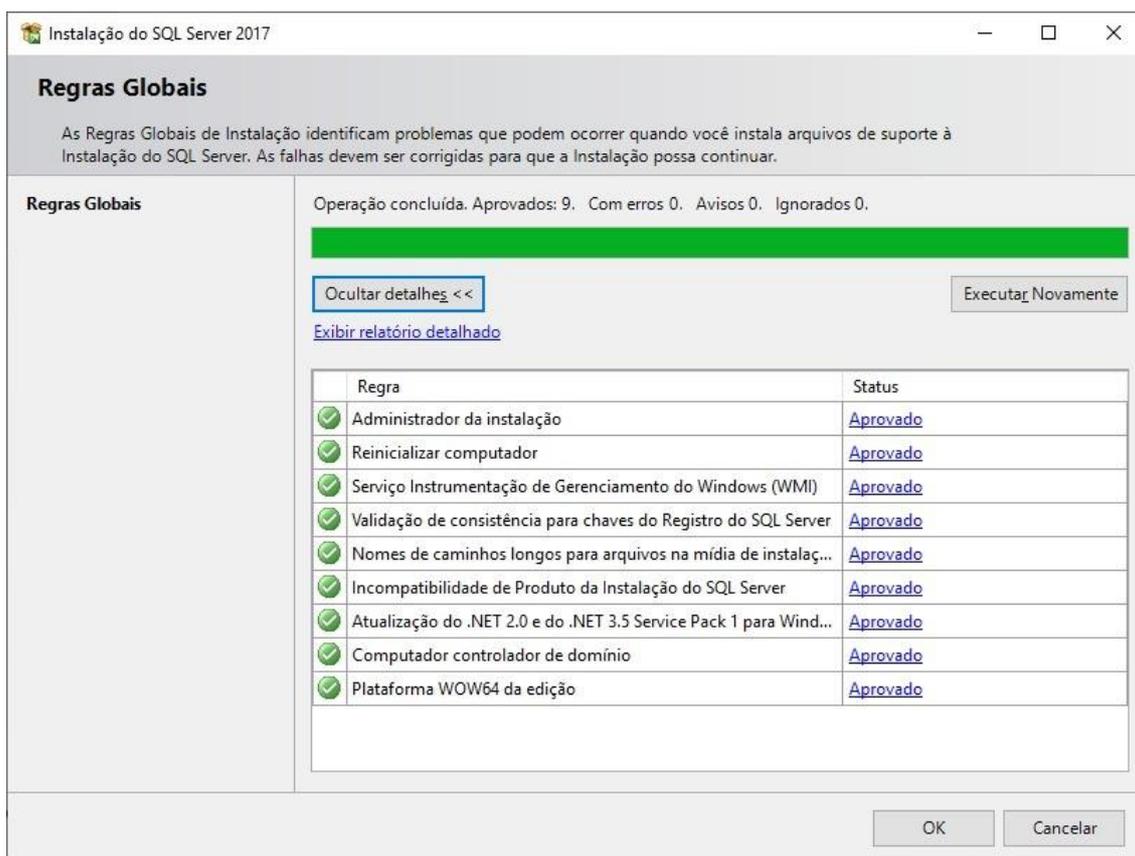
Avançado: Contém uma opção para instalar o SQL Server usando um arquivo de configuração, assim como opções de cluster avançadas

Opções: Permite especificar o diretório-raiz da mídia de SQL Server

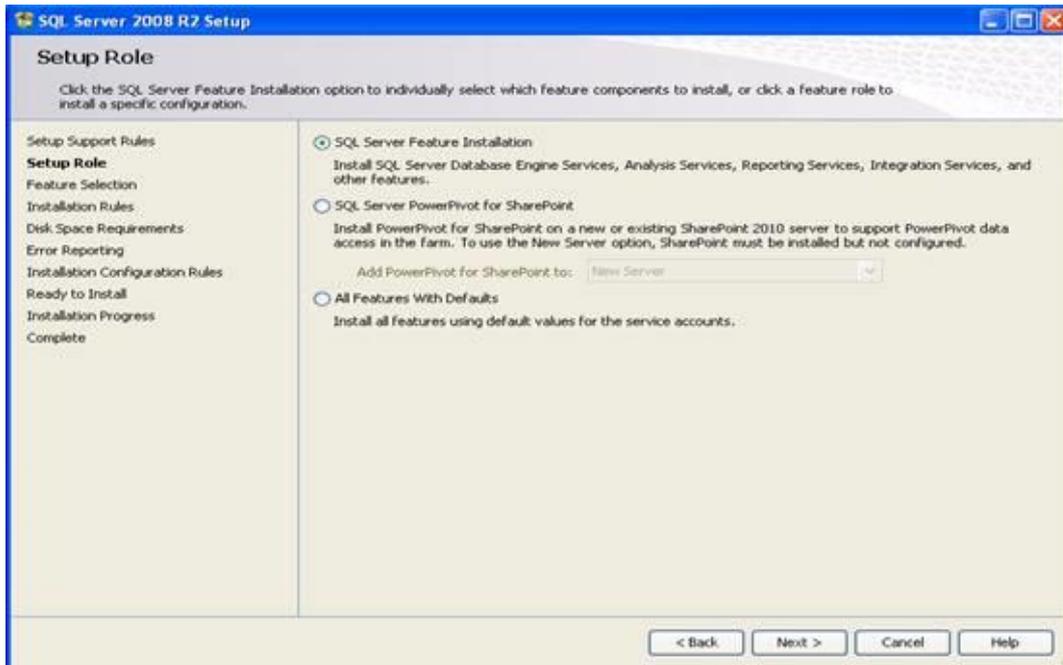
A próxima tela do Setup do SQL Server examina a máquina onde o SQL Server será instalado.

Aparecerá a tela que informa os termos de uso do SQL Server. Marque a opção "I accept the license terms" e clique em "Next".

SQL Server Setup verifica se existe alguma pendência com relação a algum software ou requisitos:



Caso exista, corrija as falhas. Se não houver, o SQL Server exibe a próxima tela:

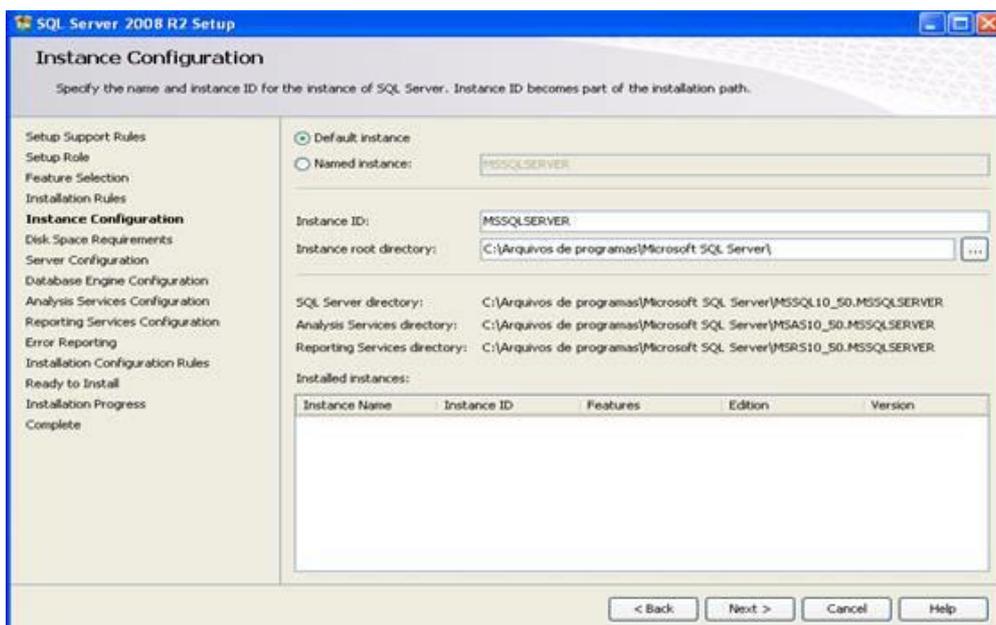


No nosso exemplo, apenas marcamos a opção “SQL Server Feature Installation” e seguimos.

O próximo passo é a escolha dos componentes do SQL Server. Nesta etapa, você deve aguardar o preenchimento da barra de status da execução.

Ao final, a verificação de compatibilidade do ambiente estará concluída e você deve continuar.

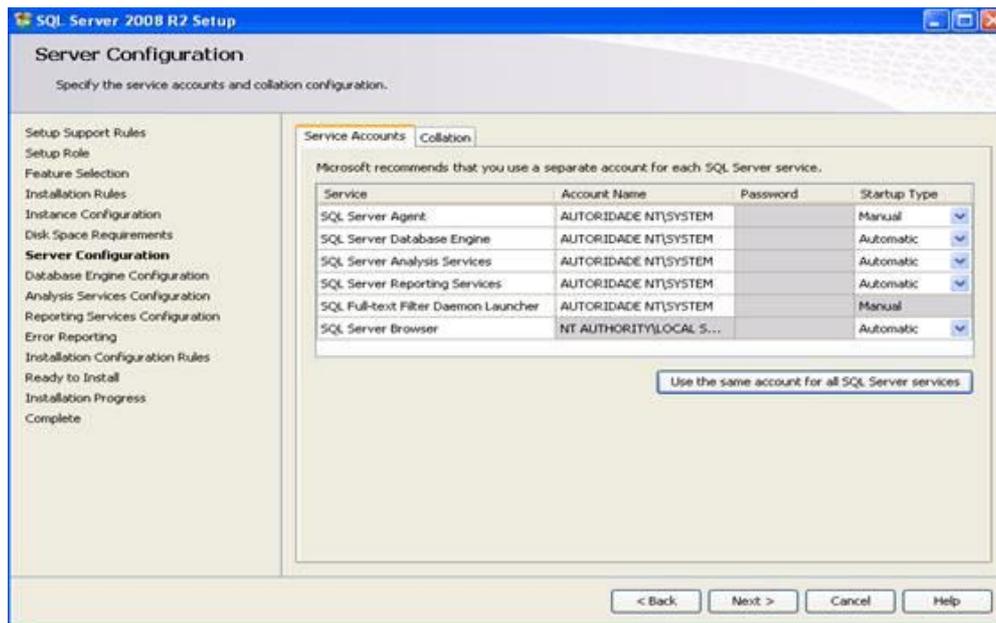
A próxima janela da instalação será Instance Configuration e você irá fazer a instalação da instância do SQL Server:



O SQL Server dá suporte a até 50 instâncias do Mecanismo de Banco de Dados em um único computador.

Aparecerá a janela “Disk Space Requirements”. Nela estará somente algumas informações sobre tamanho e local onde os arquivos estarão localizados.

A próxima janela “Server Configuration” requer um pouco mais de cuidado. A janela vem com opções de login para serviços do SQL Server:



Próxima janela é a “Database Engine Configuration”. Você vai configurar uma instância de Database Engine.

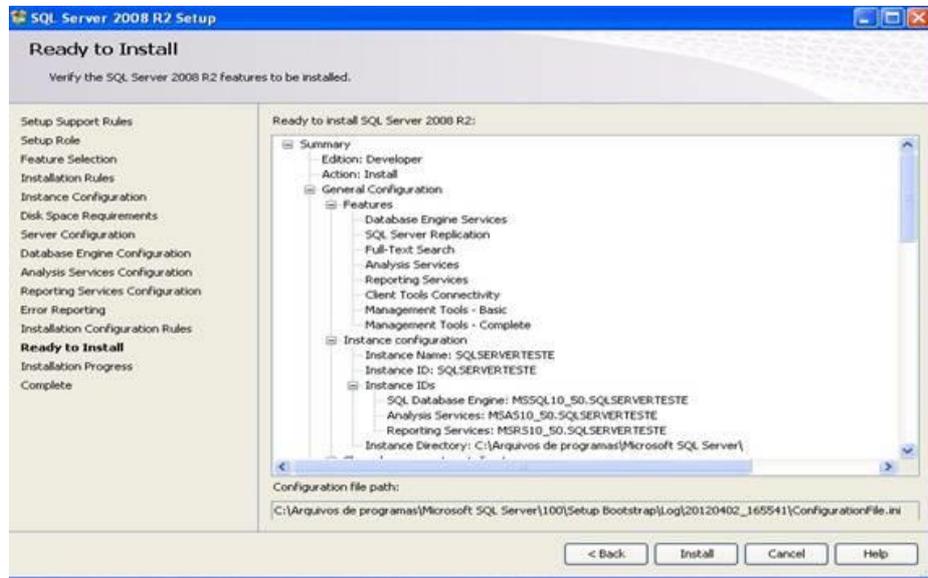
A próxima janela “Analysis Services Configuration” aparecerá sempre que você marcar no “Feature Selection” a opção Analysis Services.

Em seguida, na janela “Reporting Services Configuration”, marque a opção “Install the native mode default configuration” e depois clique em “Next”.

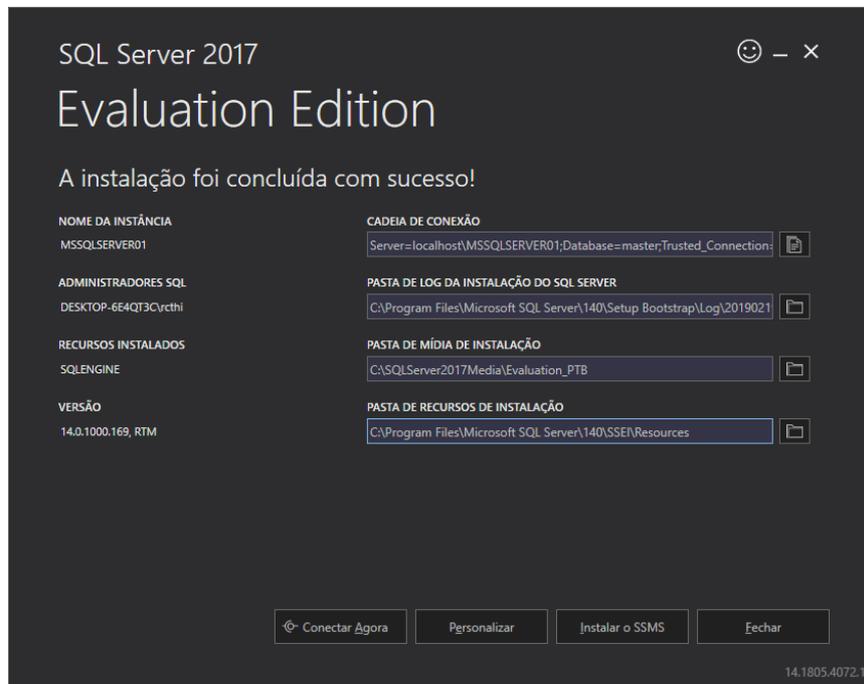
Primeiramente, é feita a instalação de roles (papéis) internos do SQL Server. Você deverá aguardar alguns minutos até a conclusão e depois clicar em “Next”.

Antes de iniciar a instalação, o setup do SQL Server te dá um relatório de tudo que será instalado.





O processo de instalação, então, começa. Aguarde alguns minutos e a instalação será concluída. Clique em “Close” e pronto!



Para confirmar se a instalação foi bem-sucedida, vá até o menu iniciar> todos os programas> Microsoft SQL Server.

Clique em SQL Server Management Studio, aparecerá a figura abaixo (aguarde até abrir o programa):





Vamos fazer a primeira questão, a qual aborda a instalação.

1. Analista Judiciário (STF)/Apoio Especializado/Suporte em Tecnologia da Informação/2013

A respeito de configuração e administração de bancos de dados, julgue o item a seguir.

No processo de instalação do SQL Server 2012, deve-se instalar apenas uma cópia das ferramentas de gerenciamento, independentemente da quantidade de instâncias do SQL Server instaladas na máquina.

Certo

Errado

Comentário: O SQL Server dá suporte a até 50 instâncias do Mecanismo de Banco de Dados em um único computador. Independentemente da quantidade de instâncias do SQL Server instaladas na máquina, deve-se instalar apenas uma cópia das ferramentas de gerenciamento. Essa cópia é suficiente para o gerenciamento das instâncias.

Gabarito: C



CONCEITOS SOBRE O DATABASE ENGINE

Inicialmente, veremos quais são os componentes do servidor e suas principais características:

| Componentes de servidor | Descrição |
|---|---|
| Mecanismo de Banco de Dados do SQL Server | Mecanismo de Banco de Dados do SQL Server inclui Mecanismo de Banco de Dados, o serviço principal para armazenamento, processamento e proteção de dados, replicação, pesquisa de texto completo, ferramentas para gerenciar dados XML e relacionais, na integração da análise de banco de dados e na integração do PolyBase para acesso ao Hadoop e a outras fontes de dados heterogêneas, bem como o servidor Data Quality Services (DQS). |
| Analysis Services | Analysis Services inclui as ferramentas para criação e gerenciamento de aplicativos OLAP (processamento analítico online) e de mineração de dados. |
| Reporting Services | O Reporting Services inclui componentes de servidor e cliente por criar, gerenciar e implantar relatórios tabulares, de matriz, gráficos e de forma livre. O Reporting Services também é uma plataforma extensível que você pode usar para desenvolver aplicativos de relatório. |
| Integration Services | Integration Services é um conjunto de ferramentas gráficas e objetos programáveis para mover, copiar e transformar dados. Ele também inclui o componente Data Quality Services (DQS) para o Integration Services. |
| Master Data Services | O Master Data Services (MDS) é a solução do SQL Server para gerenciamento de dados mestre. O MDS pode ser configurado para gerenciar qualquer domínio (produtos, clientes, contas) e inclui hierarquias, segurança granular, transações, controle de versão de dados e regras de negócio, bem como um Suplemento para Excel que pode ser usado para gerenciar dados. |
| Serviços de Machine Learning (No Banco de Dados) | Os Serviços de Machine Learning (No Banco de Dados) oferecem suporte a soluções escalonáveis de aprendizado de máquina por meio de fontes de dados empresariais. No SQL Server 2016, havia suporte para a linguagem R. O SQL Server 2017 oferece suporte às linguagens R e Python. |



Servidor do Machine Learning (Autônomo)

O Servidor do Machine Learning (Autônomo) oferece suporte à implantação de soluções de aprendizado de máquina distribuídas e escalonáveis em várias plataformas, usando várias fontes de dados empresariais, inclusive Linux e Hadoop. No SQL Server 2016, havia suporte para a linguagem R. O SQL Server 2017 oferece suporte às linguagens R e Python.

O Mecanismo de banco de dados (Database Engine) é o cerne dos componentes do SQL Server. O motor funciona como um serviço em uma máquina, que é muitas vezes referido como uma instância do SQL Server. É possível executar várias instâncias do SQL Server em um determinado servidor. Quando você acessa o SQL Server, a instância recebe uma conexão. Uma vez que um aplicativo é conectado, ele envia declarações de Transact-SQL (T-SQL) à instância.

A instância por sua vez envia dados de volta para o cliente. Dentro da conexão, uma camada de segurança valida o acesso aos dados, conforme especificado pelos administradores de banco de dados (DBAs). A Database Engine permite que você aproveite todos os recursos dos outros componentes, como acesso, armazenamento e proteção dos dados.

O componente de armazenamento da Database Engine determina como os dados são armazenados no disco. Ao projetar seus bancos de dados, você vai especificar vários aspectos que irão ditar como suas tabelas, índices e, em alguns casos, visões são fisicamente organizados em seu subsistema de disco. No SQL Server, você pode distribuir fisicamente dados em discos dividindo-o, ou separando em partes distintas e independentes, essa função é conhecida como *partitioning* ou particionamento.

O particionamento não só melhora o desempenho das consultas, mas também simplifica o processo de gerenciamento e manutenção de seus dados. Com o lançamento do SQL Server 2012, a Microsoft aumentou o número de partições suportadas para 15.000 por tabela.

Dentro da própria Database Engine, o mecanismo de armazenamento é o componente primário. Circundantes a ele estão vários componentes adicionais que dependem do motor. Estes componentes incluem:

- Interface para programação T-SQL (implementação da Microsoft da linguagem padrão SQL ANSI)
- Subsistema de segurança
- Replicação
- SQL Server Agent
- Ferramentas para alta disponibilidade e recuperação
- SQL Server Integration Services
- Ferramentas de gerenciamento do SQL Server

As seções a seguir fornecem uma breve explicação sobre cada componente.



INTERFACE PARA PROGRAMAÇÃO T-SQL

O SQL Server fornece uma linguagem de programação que permite escrever consultas simples e complexas a serem executadas sobre as estruturas de armazenamento. Usando T-SQL, você pode escrever consultas de manipulação de dados que permitem modificar e acessar os dados sobre demanda.

Você pode criar objetos como visões, procedimentos armazenados, gatilhos e funções definidas pelo usuário que agem como um meio de acessar esses dados. As aplicações escritas em linguagens de programação tais como Visual Basic e C# .NET podem enviar consultas T-SQL das aplicações para o Database Engine.

A Database Engine, então, processa as consultas e envia os resultados de volta para o cliente. Além disso, você pode escrever comandos de definição de dados (DDL) para criar ou modificar objetos que atuam como mecanismos para armazenar os dados. T-SQL também permite gerenciar as configurações e a segurança de servidor.

T-SQL é uma linguagem baseada em conjunto, o que significa que ele executa de forma ideal as operações com conjuntos em oposição à manipulação de strings ou iteração sobre linhas de dados. T-SQL é capaz de executar operações baseados em cursores, contudo esses tipos de operações são menos eficientes do que uma operação de conjunto.

Se você achar que está usando T-SQL para executar operações baseados em cursores, considere o uso de uma linguagem Common Language Runtime (CLR). Usando seu compilador favorito (Visual Studio, por exemplo), assim você pode estender as funcionalidades do T-SQL e obter o melhor desempenho das duas linguagens.

SQL Server 2012 apresenta vários novos aprimoramentos de programação T-SQL, incluindo uma forma simples de paginação, funções de janelas e tratamento de erros. A instrução *throw* que foi introduzido e fornece uma maneira de lidar elegantemente com erros lançando exceções. Agora você pode criar um arquivo de tabela que se baseia na tecnologia File Stream introduzido no SQL Server 2008. O acoplamento do FileTable com fulltextsearch permite executar consultas complicadas contra grandes quantidades de dados de texto. SQL Server 2012 também introduz várias novas conversões, e funções para manipulação de strings, date e time.

SUBSISTEMA DE SEGURANÇA

Na maioria das organizações, os dados são o bem mais valioso, e manter os dados seguros é uma grande preocupação. Qualquer vulnerabilidade na segurança dos dados pode acabar provocando uma série de eventos que podem se revelar catastróficos para o negócio. É por isso que SQL Server 2012 consiste de um robusto subsistema de segurança que permite o controle de acesso através de dois modos de autenticação, SQL e Windows.

Como administrador, você é capaz de configurar a segurança do SQL Server em vários níveis. Usando T-SQL ou SQL Server Management Studio, você pode controlar o acesso a uma determinada



instância do SQL Server a uma base de dados específica, para objetos dentro desses bancos de dados, e até mesmo para colunas dentro de uma tabela particular.

SQL Server também inclui criptografia nativa. Por exemplo, se você quiser proteger o CPF e salário dos empregados, é possível usar a criptografia em nível de coluna, você pode criptografar apenas uma única coluna em uma tabela. SQL Server possui ainda uma criptografia de dados transparente (TDE), que permite criptografar todo um banco de dados sem afetar a forma como os clientes e aplicações acessam os dados. No entanto, se alguém violar a segurança da rede e obter uma cópia de um arquivo de dados ou arquivo de backup, a única maneira que da pessoa acessar os dados é por meio da chave de criptografia que você definiu e armazenou.

Mesmo com todos esses recursos de segurança, o SQL Server oferece a você a capacidade de auditar o seu servidor de bancos de dados de forma proativa. Em SQL Server 2012, você pode filtrar eventos de auditoria antes de serem escritas no log de auditoria.

Também em SQL Server 2012, você pode criar ROLES de servidor definidas pelo usuário, que podem ajudar na criação de um método mais seguro de alocação de acesso para administradores de servidores. Microsoft incluiu a capacidade de criar usuários dentro de um banco de dados sem a necessidade de criar um login no servidor, conhecido como bancos de dados autocontidos. Em versões anteriores do SQL Server, antes de conceder acesso no nível de banco de dados, um administrador precisava criar um login do servidor. Com o advento do SQL Server 2012, um usuário pode ser autocontido dentro de um banco de dados.

1. Analista do Ministério Público de Sergipe/Informática I/Gestão e Análise de Projeto de Infraestrutura/2010

Um ambiente integrado para acessar, configurar, gerenciar, administrar e desenvolver todos os componentes do SQL Server 2005 é o

- a) Microsoft Office System.
- b) Tools and Utilities Documentation.
- c) Microsoft SQL Server Management Studio.
- d) Business Intelligence Development Studio.
- e) Microsoft Visual Studio.

Comentário: SQL Server Management Studio (SSMS) é um ambiente integrado para acessar, configurar, gerenciar, administrar e desenvolver todos os componentes do SQL Server. O SSMS combina um amplo grupo de ferramentas gráficas com vários editores de script avançados para fornecer acesso ao SQL Server para desenvolvedores e administradores de todos os níveis de conhecimento.

O SSMS combina os recursos do Enterprise Manager, do Analisador de Consultas e do Analysis Manager, incluídos em versões anteriores do SQL Server, em um único ambiente. Além disso, o SSMS funciona com todos os componentes do SQL Server, tais como o Reporting Services e o Integration Services. Desenvolvedores terão uma experiência familiar e os administradores



de banco de dados terão um único utilitário abrangente que combina ferramentas gráficas fáceis de usar com sofisticadas capacidades de script.

Assim, temos o gabarito na letra c).

Gabarito: C

REPLICAÇÃO

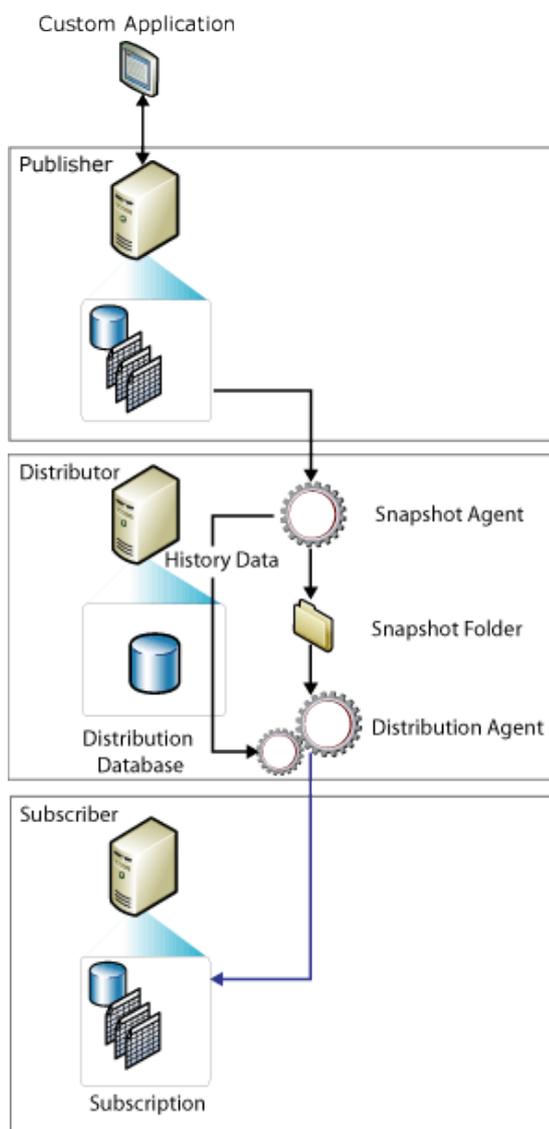


Figura 1 - Replicação de instantâneo ou snapshot

A replicação do SQL Server está disponível na maioria das versões do produto. Com o tempo, diferentes tipos de replicação foram introduzidos para garantir que os usuários possam configurar suas arquiteturas de replicação. Isso garante que a replicação pode satisfazer uma ampla variedade de cenários. Usando tecnologia de replicação do SQL Server, você pode distribuir seus dados, em diferentes locais, utilizando File Transfer Protocol (FTP), através da Internet, e para os usuários móveis. A replicação pode ser configurada para enviar, extrair e mesclar dados através das redes locais (LANs) e redes de área (WANs).

A forma mais simples de replicação, replicação de instantâneo ou snapshot, periodicamente é feita uma fotografia dos dados e, os mesmos são distribuídos aos servidores que estão registrados na publicação. A replicação de instantâneo é normalmente usada para mover os dados em intervalos mais longos, como uma transferência diurna ou noturna. Embora este método seja eficaz, é muitas vezes insuficiente para satisfazer a alta demanda de usuários por dados quase em tempo real.

Se o tempo necessário para o usuário acessar os dados atualizados nas réplicas for pequeno é recomendável usar a replicação transacional. Em vez de distribuir fotografias dos dados, a replicação transacional envia continuamente as alterações de dados como eles acontecem para os assinantes. A replicação transacional é normalmente utilizada numa topologia server-to-server, onde um servidor é a fonte dos dados e outro servidor é usado como uma cópia de backup ou para *reporting*.

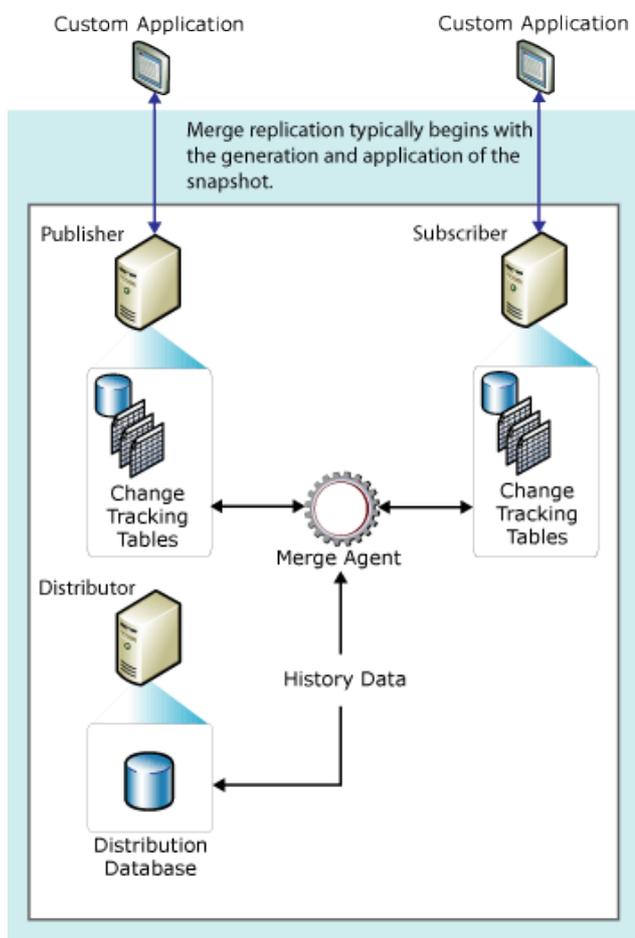


Figura 3 - Merge Replication

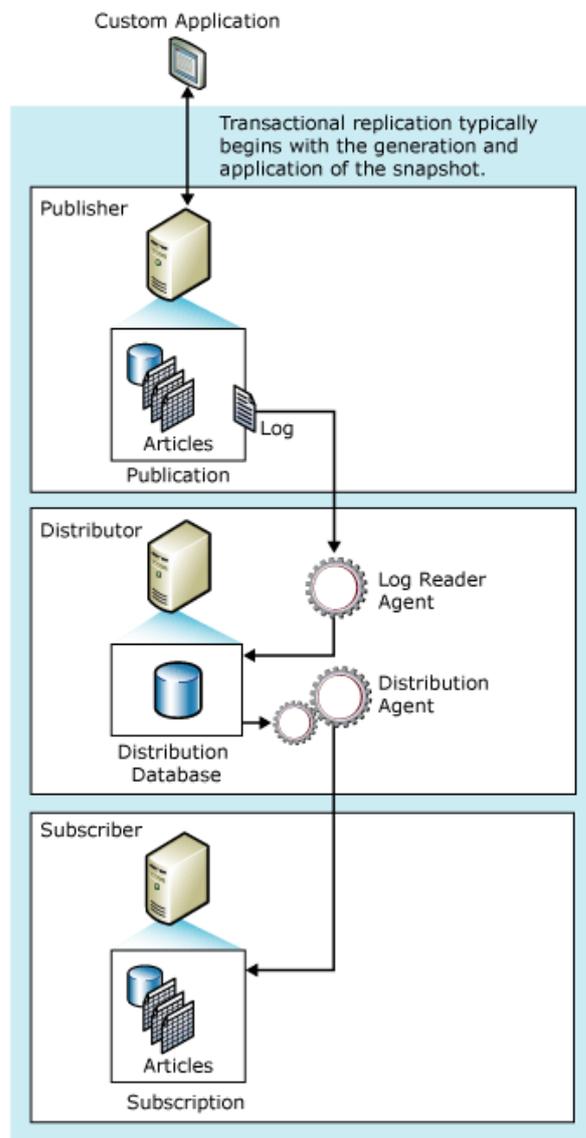


Figura 2 - Replicação transacional

Ambos os tipos de replicação de dados movimentam as informações em um único sentido. Mas, e se você precisa de uma movimentação bidirecional? Por exemplo, suponha que você tem usuários móveis que trabalham offline. Enquanto eles estiverem off-line, eles

entram informações em um banco de dados locais que residem em uma réplica da instância do SQL Server em execução em seus laptops. O que acontece quando eles retornam para o escritório e se conectam à rede?

Neste cenário, a instância local irá sincronizar com o banco de dados primário da empresa. A replicação de *merge* moverá as transações entre o *publisher* e assinante desde a última sincronização.

Profissionais do SQL Server debatem constantemente o uso da replicação como uma tecnologia para alta disponibilidade (HA) ou de recuperação de desastres (DR). Poderia ser usado para qualquer um?



Existe uma possibilidade; no entanto, a replicação única move apenas as mudanças nos esquemas e nos dados. Para fornecer uma topologia de HA ou DR eficaz, cada aspecto da instância deve ser incluído, tais como segurança, ações de manutenção, jobs, e assim por diante. Portanto, usando a replicação em ambos os casos pode representar problemas potenciais em caso de falha de hardware ou de um desastre.

SQL SERVER AGENT

O SQL Server Agent é executado como um serviço separado em uma instância do SQL Server. Cada instância do SQL Server tem um serviço SQL Agent. O principal uso do SQL Server Agent para executar tarefas agendadas, como a reconstrução de índices, o backup de bancos de dados, o carregamento (load) do armazém de dados, e assim por diante.

Ele permite que você agende as tarefas para executar em vários intervalos ao longo do dia ou da noite. Para garantir que você será notificado em caso de uma falha, o SQL Server Agent permite que você configure operadores e alertas. Um operador é simplesmente um indivíduo e um endereço de e-mail. Uma vez que você configure um operador, você pode enviar notificações ou alertas a essa pessoa quando um trabalho termina com sucesso, completa suas ações, ou falha.

HIGH AVAILABILITY AND DISASTER RECOVERY TOOLS

Com demandas crescentes sobre a disponibilidade e o tempo de resposta do servidor, é vital que os SGBDs incluam vários mecanismos que garantam a coerência e a disponibilidade de seus dados. SQL Server 2012 fornece quatro tecnologias para alta disponibilidade:

Grupos de Disponibilidade AlwaysOn - No SQL Server 2012, a Microsoft introduz os Grupos de Disponibilidade AlwaysOn. Um grupo de disponibilidade suporta failover para um conjunto de bancos de dados e melhora a tecnologia de espelhamento base de dados existente para manter réplicas secundárias de base de dados em *snapshots* locais ou remotos. Esta tecnologia difere de failover tradicional *clustering* de duas maneiras:

- Você pode configurar o *failover* automático sem o uso de um Storage Area Network (SAN).
- Você pode configurar uma ou mais das réplicas secundárias para apoiar operações somente leitura.

Uma vez que uma rede SAN não é mais necessária, agora você tem a capacidade de configurar HA e DR. Ao evoluir a capacidade de espelhamento do banco de dados ele permite mover dados por longas distâncias usando TCP/IP. Você pode ter uma cópia do banco de dados armazenados em um database center localizado em uma área geográfica diferente.

Failover Clustering – As instancias de failover cluster do SQL Server fornecem suporte de alta disponibilidade no nível do servidor. Antes de construir uma instância de failover do servidor, você deve criar e configurar um cluster de failover do Windows Server.

Database Mirroring - Antecessor do AlwaysOn, o espelhamento de banco de dados fornece alta disponibilidade no nível do banco de dados. Ele mantém duas cópias do banco de dados em



instâncias do SQL Server rodando em servidores separados. Normalmente, os servidores estão hospedados em locais geográficos distintos, não só assegurando HA, mas também fornecendo DR. Se você quiser incorporar o failover automático, você deve incluir um terceiro servidor (testemunha) que vai mudar qual servidor é o proprietário do banco de dados.

Ao contrário do AlwaysOn, com banco de dados espelhamento você não pode ler diretamente da cópia secundária. Você pode, no entanto, criar um snapshot para fins de leitura somente. Esse instantâneo terá um nome diferente, por isso quaisquer clientes que se conectam a ele devem estar cientes da mudança de nome. Note que este recurso foi preterido e substituído pelo AlwaysOn que deve ser usado a partir de agora em vez do espelhamento de banco de dados.

Log Shipping - Esta é outra tecnologia que oferece alta disponibilidade no nível de banco de dados, que é ideal para redes com latência muito baixa. O log de transações para um banco de dados específico é enviado para um servidor secundário a partir do servidor primário e restaurado. Assim como com AlwaysOn e banco de dados espelhamento, você pode configurar o envio de log de uma forma que permite que o banco de dados secundário seja lido.

Nota: Se você estiver familiarizado com o SQL Server, você pode estar se perguntando por que a replicação não aparece na lista anterior. Isso ocorre porque a replicação carece de algumas características-chaves, tais como sincronização de dados holística (em oposição a movimentações apenas de objetos).

SQL SERVER INTEGRATION SERVICES

SQL Server Integration Services (SSIS) é uma plataforma que permite que você construa extração, transformação e carga (ETL) de alto desempenho para estruturas de data warehouses. Então, por que é incluído no aqui em uma lista de componentes de motor de banco de dados? Na maioria dos casos o SSIS é usado para ETL, no entanto, ela oferece um número de tarefas e transformações que se estendem bem além de seu uso ETL.

Por exemplo, se você é novo para a administração de um ambiente SQL Server, SSIS fornece as ferramentas necessárias para executar várias tarefas administrativas, incluindo a reconstrução de índices, atualizações estatísticas, e backup de bancos de dados, que compõem a lista preliminar de itens de manutenção que deve ser realizada em qualquer base de dados. Sem o SSIS, como um novo administrador que você poderia gastar muito tempo escrevendo comandos T-SQL apenas para colocar essas atividades em execução.

Mas estas não são as únicas funcionalidades de SSIS para os administradores. Quantas vezes você foi solicitado para fazer uma exportação de dados para Microsoft Excel ou para mover dados de um servidor para outro? Usando SSIS, você pode rapidamente exportar ou importar dados a partir de várias fontes, incluindo Excel, arquivos de texto, Oracle e DB2.

FERRAMENTAS DE GERENCIAMENTO SQL SERVER

O SQL Server 2012 inclui duas interfaces gráficas que permitem gerenciar, monitorar, manter, e desenvolver em um ambiente SQL Server. O primeiro é o SQL Server Management Studio (SSMS),



que permite executar praticamente qualquer ação que você pode pensar contra uma instância do SQL Server. É um ambiente integrado onde você pode acessar muitas instâncias do SQL Server. É constituído por um amplo conjunto de ferramentas com um rico conjunto de interfaces gráficas e editores de script que simplificam o processo de desenvolvimento e configuração das instâncias do SQL Server.

Além de SSMS, SQL Server 2012 introduz o SQL Server Data Tools (SSDT). O SSDT é outro ambiente integrado, mas que foi projetado especificamente para desenvolvedores de banco de dados. Você pode explorar o banco de dados e seus objetos usando o SQL Server Object Explorer. Alguns dos recursos mais interessantes do SSDT são a capacidade de criar ou editar objetos e a forma de execução de consultas diretamente na interface. Usando a interface visual para criação de tabelas (Table Designer), você pode alterar esquemas de tabela para ambos os projetos de bancos de dados e instâncias de banco de dados on-line.

O SQL Server disponibiliza também o SQL Server Configuration Manager. Esta ferramenta deve ser usada para iniciar, pausar, retomar ou interromper os serviços, para exibir as propriedades de serviço ou para alterar as propriedades de serviço. O SQL Server Configuration Manager também deve ser usado para iniciar o Mecanismo de Banco de Dados usando parâmetros de inicialização.

2. Auditor de Controle Externo (TCE-PA)/Informática/Analista de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

A ferramenta SQL Server Configuration Manager permite realizar configurações de modo que uma instância do SQL Server se inicie automaticamente quando o servidor for ligado.

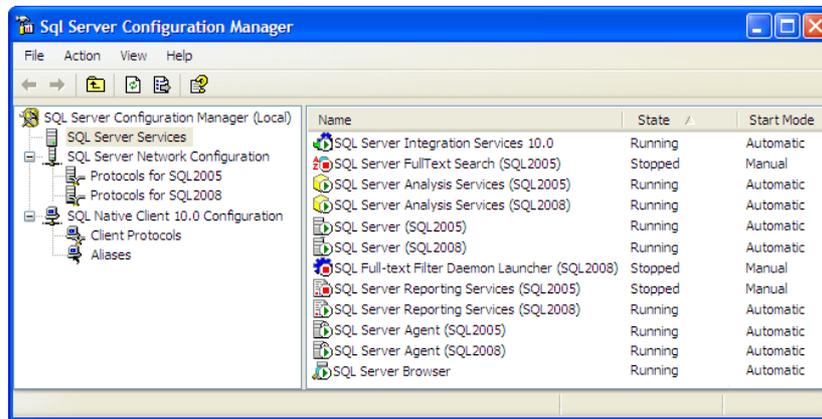
Certo

Errado

Comentário: O gabarito é correto, pois o SQL Server Configuration Manager é uma ferramenta para gerenciar os serviços associados ao SQL Server, configurar os protocolos de rede usados pelo SQL Server e para gerenciar a configuração de conectividade de rede de computadores cliente do SQL Server.

Durante a instalação, o SQL Server normalmente é configurado para que a instância do SQL Server inicie automaticamente. Se isto não foi feito, você poderá alterar essa definição a qualquer momento.





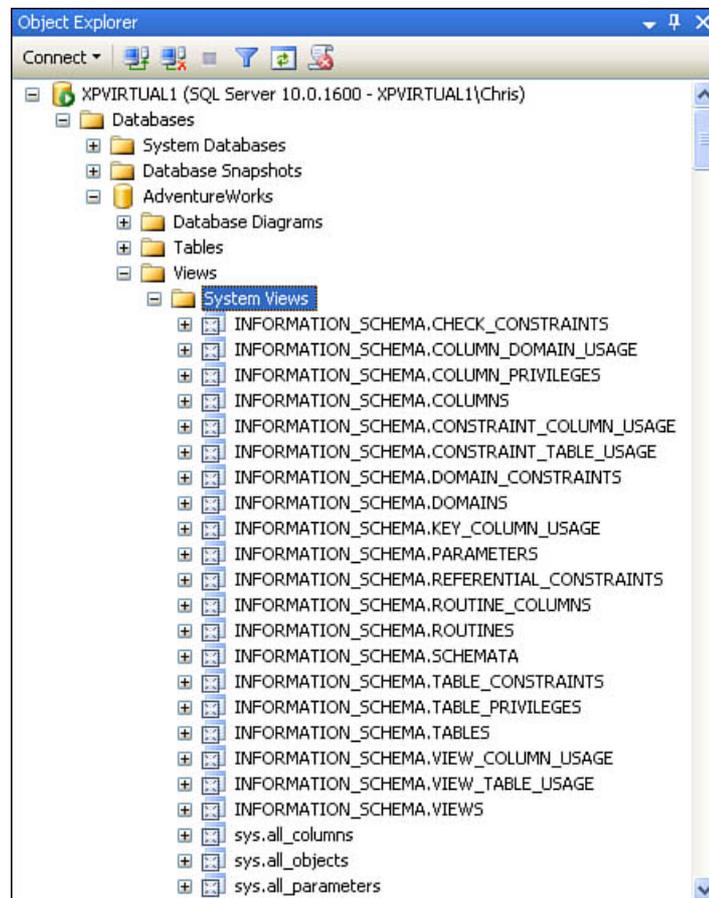
Gabarito: C

CATÁLOGO INTERNO (VIEWS SYS.*)

System views são tabelas virtuais que expõem metadados que dizem respeito a vários aspectos diferentes do SQL Server. Vários tipos diferentes de visões dão suporte a diferentes necessidades de dados. O SQL Server 2008 oferece um vasto número de visões de sistema em seu catálogo interno que devem atender a maioria, se não todas, as suas necessidades de metadados.

As visões de sistema disponíveis podem ser observadas no Object Explorer do SSMS. A figura a seguir mostra a ferramenta com o nó referente a visões de sistema em destaque. São várias visões para cobrirmos em detalhe nesta aula. Tentaremos fornecer alguns exemplos para mostrar o valor ou a relevância destes objetos. Cada visão do sistema é coberta em detalhes no SQL Server Books Online, que inclui descrições de cada coluna.





As visões do INFORMATION_SCHEMA fornecem outra opção independente de tabela do sistema para acessar metadados do SQL Server. Este tipo de visão estava disponível em versões anteriores do SQL Server. Usar esse esquema de informações é uma alternativa viável para acessar metadados SQL Server a partir de um aplicativo de produção. As visões do esquema de informações habilitam um aplicativo que as usa para funcionar corretamente, embora as tabelas do sistema subjacente possam ter mudado. Alterações nas tabelas do sistema subjacente são mais prevalentes quando uma nova versão do SQL Server é liberada (como o SQL Server 2008), mas as mudanças também podem ocorrer como parte de pacotes de serviços na versão existente.

As views do esquema de informações também têm a vantagem de ser compatível com SQL-92. Essa conformidade com o padrão SQL-92 significa que instruções SQL escritas contra essas visões funcionam com outros SGBDs, que também aderem ao padrão SQL-92. O padrão SQL-92 suporta uma convenção de nomenclatura de três partes, que foi implementado no SQL Server, qual seja, database.schema.object.

No SQL Server 2008, todas as visões do esquema de informações estão no mesmo esquema, chamado INFORMATION_SCHEMA. As seguintes views ou objetos de esquema de informação estão disponíveis:

- CHECK_CONSTRAINTS
- COLUMN_DOMAIN_USAGE
- COLUMN_PRIVILEGES
- COLUMNS



- CONSTRAINT_COLUMN_USAGE
- CONSTRAINT_TABLE_USAGE
- DOMAIN_CONSTRAINTS
- DOMAINS
- KEY_COLUMN_USAGE
- PARAMETERS
- REFERENTIAL_CONSTRAINTS
- ROUTINES
- ROUTINE_COLUMNS
- SCHEMATA
- TABLE_CONSTRAINTS
- TABLE_PRIVILEGES
- TABLES
- VIEW_COLUMN_USAGE
- VIEW_TABLE_USAGE
- VIEWS

Ao se referir a uma visão do esquema de informações em uma instrução SQL, você deve usar um nome qualificado que inclui o nome do esquema. Por exemplo, a seguinte instrução retorna todas as tabelas e colunas em um banco de dados, utilizando as tabelas e colunas da visão do esquema de informações:

```
select t.TABLE_NAME, c.COLUMN_NAME
from INFORMATION_SCHEMA.TABLES t
join INFORMATION_SCHEMA.COLUMNS c on t.TABLE_NAME = c.TABLE_NAME
order by t.TABLE_NAME, ORDINAL_POSITION
```

PROJETANDO UM BANCO DE DADOS SQL SERVER

O banco de dados é o contêiner para todos os objetos dentro Microsoft SQL Server para o motor relacional. Nesta parte da aula, você vai aprender sobre os bancos de dados de sistema que armazenam informações vitais sobre a instância do SQL Server.

Vamos aprender também técnicas fundamentais para criar bancos de dados, juntamente com métodos para controlar como e onde os dados são armazenados. Os métodos incluem a criação de bases de dados que consistem em vários grupos de arquivos (filegroups) e vários arquivos de dados. Finalmente, você vai aprender como mover um banco de dados de uma instância do SQL Server para outra, e explorar os modelos de recuperação de banco de dados.

BASES DE DADOS DE SISTEMA

Antes de começarmos a criar bancos de dados Microsoft SQL Server, você deve ter uma boa compreensão das bases de dados do sistema que são criadas por padrão quando você instalar uma instância do SQL Server. Cada um dos bancos de dados serve a um propósito específico e é necessário para executar o SQL Server:



master - O banco de dados mestre, como seu nome sugere, é o banco de dados primário do sistema. Sem ele, o SQL Server não pode iniciar. O banco de dados mestre contém as informações mais importantes sobre objetos dentro da instância do SQL Server, como o seguinte: Databases, AlwaysON, Database mirroring, Configurations, Logins, Resource Governor e Endpoints

tempdb - O banco de dados tempdb é um playground global para objetos temporários criados pelos processos internos que executam no SQL Server e objetos temporários que são criados por usuários ou aplicativos. Esses objetos incluem tabelas temporárias e procedimentos armazenados, variáveis de tabela, tabelas temporárias globais, e cursores. Além de objetos temporários, o tempdb armazena versões de linha para transações read-committed ou snapshot, operações de índice online, e AFTER triggers. Uma coisa importante para se notar sobre tempdb é que ele é recriado sempre que o SQL Server for reiniciado. Embora você possa criar objetos em tempdb, você nunca deve usá-lo como um banco de dados onde a informação armazenada é persistida.

model - O banco de dados modelo é exatamente o que o nome indica: um modelo para todos os bancos de dados criados em uma instância do SQL Server. Em outras palavras, ele é usado como um modelo cada vez que você cria um banco de dados. Por exemplo, se você quiser uma tabela específica de exista em cada banco de dados criado na instância do SQL Server, você irá criar essa tabela no banco de dados modelo. Como resultado, cada vez que um banco de dados é criado, ele irá incluir essa tabela.

msdb - Esta base de dados serve principalmente como o banco de dados back-end para o Microsoft SQL Server Agent. Sempre que você criar e/ou agendar um trabalho do SQL Server Agent, os metadados para esse trabalho são armazenados neste banco de dados. Além de dados SQL Server Agent, o msdb armazena informações para os seguintes componentes: Service brokers, Alerts, Log shipping, SSIS packages, Utility control point (UCP), Database mail e Maintenance plans.

resource - O banco de dados de recursos é, um banco de dados oculto somente leitura que geralmente não é discutido com muita frequência. O principal objetivo dele é o de melhorar o processo de atualização de uma versão do SQL Server para a próxima. Todos os objetos do sistema de uma instância do SQL Server são armazenados no banco de dados de recursos. Este banco de dados não pode ser copiado ou restaurado. Você não deve tentar alterar ou movimento esta base de dados, a menos que o Suporte ao Cliente Microsoft o oriente para fazer isso.

Distribution – Nossa última base de dados de sistema é a base de dados de distribuição. Esta base de dados só existe quando você tiver configurado sua instância como um distribuidor para replicação. Antes de configurar a replicação, você deve executar essa configuração. Todos os metadados e histórico para os vários tipos de replicação são armazenados dentro desta base de dados.

Vamos agora resolver uma questão de provas passadas:

3. Técnico Judiciário (TRT 9ª Região)/Apoio Especializado/Tecnologia da Informação/2015



A automatização de administração em várias instâncias do banco de dados SQL Server é chamada administração multisservidor. A administração multisservidor é utilizada

- a) exclusivamente para gerenciar dois ou mais servidores.
- b) como o único recurso oferecido pelo SQL Server para gerenciamento de servidores.
- c) somente se houver pelo menos um servidor mestre e três servidores de destino.
- d) também para programar fluxos de informações entre servidores corporativos para data warehousing.
- e) para membros da função sysadmin do servidor de destino editarem as operações que são executadas no servidor de destino pelo servidor mestre.

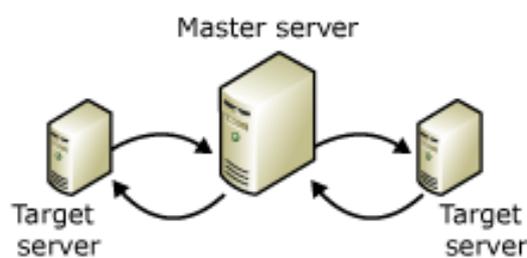
Comentário: A automatização de administração em várias instâncias de SQL Server é chamada administração multisservidor. Use a administração multisservidor para fazer o seguinte:

- Gerenciar dois ou mais servidores.
- Programar fluxos de informações entre servidores corporativos para data warehousing.

Para aproveitar a administração multisservidores, é necessário ter pelo menos um servidor mestre e um servidor de destino. Um servidor mestre distribui trabalhos para servidores de destino e também recebe eventos desses servidores. Ele também armazena a cópia central das definições de trabalho para trabalhos que são executados em servidores de destino. Os servidores de destino conectam-se periodicamente ao servidor mestre para atualizar a agenda de trabalhos.

A ilustração a seguir mostra a relação entre servidores mestre e de destino:

Multiserver administration configuration



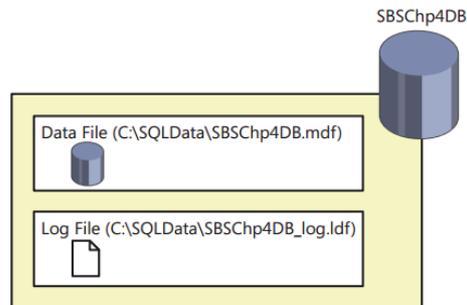
Gabarito: D

ESTRUTURA DA BASE DE DADOS

Os bancos de dados são os objetos primários de armazenamento de dados dentro do SQL Server. O processo de criação do banco de dados, embora muito simples, requer uma reflexão cuidadosa sobre a sua estrutura. Bancos de dados podem ser criados usando várias tecnologias e técnicas diferentes, por exemplo T-SQL e SSMS.



Por padrão, todos os bancos de dados SQL Server consistem de dois tipos de arquivos. O **arquivo de dados** contém os dados e objetos do banco de dados como tabelas, visões e procedimentos armazenados. O **arquivo de log** contém informações que auxiliam na recuperação de transações no banco de dados. A estrutura de banco de dados SQL Server consiste em, pelo menos, um arquivo de dados e um arquivo de log. Vejam a figura abaixo:



Existem dois tipos de arquivos de dados: primário e secundário. Quando um banco de dados é criado inicialmente, o arquivo de dados principal é criado. Por padrão, ele contém todas as informações de inicialização do banco de dados. Quando os objetos definidos pelo usuário são criados, eles também podem ser armazenados no arquivo de dados primário. No entanto, você pode definir estratégias de arquitetura que visam melhorar o desempenho, escalabilidade e facilidade de manutenção do banco de dados.

Em vez de colocar objetos definidos pelo usuário no arquivo de dados principal, você tem a opção de adicionar um arquivo de dados secundário para seu banco de dados. Estes tipos de arquivos são normalmente identificados pela extensão de arquivo: arquivos primários são normalmente seguidos de .mdf, enquanto os arquivos secundários possuem o sufixo .ndf. Não é um requisito, no entanto, é uma prática recomendada usar essas extensões. Os arquivos de dados secundários são muitas vezes utilizados para espalhar dados através de subsistemas de disco ou para adicionar mais espaço em disco para um banco de dados, no caso dos outros arquivos de dados atingirem a capacidade máxima.

4. Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

Considerando o sistema gerenciador de banco de dados Microsoft SQL Server 2008 sobre os arquivos presentes nesse gerenciador é correto afirmar:

- Um banco de dados deve conter pelo menos um arquivo de dados secundários.
- Um determinado banco de dados pode conter mais de um arquivo de log.
- A extensão recomendada para arquivos de dados primários é .ndf.
- Um determinado banco de dados pode não conter qualquer arquivo de log.
- A extensão recomendada para arquivos de dados secundários é .sdf.

Comentário: É possível obter mais espaço aumentando o arquivo de log existente (se houver espaço em disco) ou adicionando um arquivo de log ao banco de dados, geralmente em um disco diferente. Um arquivo de log de transações é suficiente, a menos que o espaço de log

esteja se esgotando e o espaço em disco também esteja se esgotando no volume que contém o arquivo de log.

Para adicionar um arquivo de log ao banco de dados, use a cláusula ADD LOG FILE da instrução ALTER DATABASE. Adicionar um arquivo de log permite o crescimento do log.

Para aumentar o arquivo de log, use a cláusula MODIFY FILE da instrução ALTER DATABASE, especificando a sintaxe SIZE e MAXSIZE.

Assim, um determinado banco de dados pode conter mais de um arquivo de log. Temos o gabarito na letra b).

Gabarito: B

CRIANDO O PRIMEIRO BANCO DE DADOS COM T-SQL

Para criar um banco de dados você pode abrir o editor de consultas do SSMS e digitar o comando abaixo:

```
--Use this script to create a database using T-SQL USE master;  
CREATE DATABASE SBSConcursosTSQL  
ON PRIMARY  
(NAME=' SBSConcursosTSQL 1', FILENAME = 'C:\SQLDATA\SBSTSQL1.mdf', SIZE=10MB,  
MAXSIZE=20, FILEGROWTH=10%)  
LOG ON (NAME=' SBSConcursosTSQL _log', FILENAME = 'C:\SQLLog\SBSTSQL_log.ldf',  
SIZE=10MB, MAXSIZE=200, FILEGROWTH=20%);
```

No script anterior, vários argumentos são usados para que o banco de dados seja colocado em um diretório específico e cresça a uma determinada taxa. O SQL Server oferece uma longa lista de argumentos que podem ampliar ainda mais os detalhes de um banco de dados quando ele é criado e onde ele é armazenado. O script anterior usa os seguintes argumentos:

- **database_name** é o nome da base de dados, que deve ser único para qualquer das bases de dados que existem na época de criação.
- **ON** especifica o grupo de arquivos e começa a seção onde o arquivo de dados está definido.
- **LOG ON** começa a seção onde o log é definido.
- **NAME** é o nome do arquivo lógico utilizado pelo SQL Server para referenciar o arquivo. Tal como acontece com database_name, deve ser único.
- **FILENAME** é a descrição do caminho no sistema operacional, acompanhado do nome arquivo, incluindo a extensão.
- **SIZE** especifica o tamanho inicial do arquivo em megabytes (MB) por padrão. Kilobytes (KB), gigabytes (GB) e terabytes (TB) também podem ser especificados.
- **MAXSIZE** especifica o tamanho máximo até onde o arquivo pode crescer (mostrado em megabytes por padrão).



- **FILEGROWTH** especifica o incremento para crescimento do arquivo caso exceda o tamanho disponível. Também é mostrado em megabytes por padrão, mas pode ser especificado como uma porcentagem.

Após a criação do banco de dados é possível removê-lo da instância do SQL utilizando o comando DROP DATABASE. Veja a sintaxe do comando abaixo:

```
-- SQL Server Syntax  
DROP DATABASE [ IF EXISTS ] { database_name | database_snapshot_name } [ ,...n ] [;]
```

MOVENDO UM BANCO DE DADOS ENTRE INSTÂNCIAS

Agora que você criou seu banco de dados, o que acontece se você precisar movê-lo para outra instância do SQL Server? Por exemplo, suponha que você deseja redistribuir o espaço livre em um servidor ou encerrar um servidor, isso exigiria que você fizesse um detach de um banco de dados de uma instância do SQL Server e, em seguida, um attach do banco de dados para sua nova instância do SQL Server. Para conseguir isso, você pode usar T-SQL ou SSMS.

Atualmente existem duas formas de attach e uma forma de detach do banco de uma instância do SQL Server. Para anexar um banco de dados, você usa o procedimento sp_attach ou o comando CREATE DATABASE especificando o argumento FOR ATTACH. Note que o procedimento armazenado do sistema sp_attach foi preterido (deprecated) e será removido das versões futuras do SQL Server. Como resultado, é recomendável que você use apenas a opção CREATE DATABASE para anexar bancos de dados.

ENTENDENDO OS MÉTODOS DE RECUPERAÇÃO

Um banco de dados SQL Server pode ser definido com um dos três modelos de recuperação:

Simple model - O modelo simples não permite backups do log de transações. Como resultado, você não pode restaurar um banco de dados para um ponto no tempo. Seu banco de dados fica vulnerável à perda de dados ao usar este modelo.

Full model - Com o modelo completo, a perda de dados é mínima quando o log de transações é apoiado em uma base de dados regular. Toda a transação está totalmente registrada no log de transações, e o log de transações continuará a crescer até que seu backup seja feito. Enquanto este modelo não adiciona sobrecarga administrativa, os seus dados estão protegidos contra a perda de dados.

Bulk-logged model - Quando você usa o modelo de log em massa, operações em massa são minimamente escritas no log, o que reduz o tamanho do log de transações. Note que este não elimina a necessidade de fazer backup do log de transações. Ao contrário do modelo de recuperação completa, no modelo de log em massa você só pode restaurar um backup até o fim; você não pode restaurar em algum ponto no tempo.



MECANISMO DE BACKUP

O componente de backup e restauração do SQL Server oferece uma proteção essencial para dados críticos armazenados em bancos de dados do SQL Server. Para minimizar o risco de perda de dados catastrófica, você precisa fazer backup dos bancos de dados para preservar as modificações feitas nos dados regularmente. Uma estratégia de backup e restauração bem-planejada ajuda a proteger bancos de dados contra perda de dados causada por várias falhas.

O SQL Server faz backup de um banco de dados completo do SQL Server para criar um backup de banco de dados ou um ou mais arquivos ou grupos de arquivos do banco de dados para criar um backup de arquivo (BACKUP DATABASE). Além disso, no modelo de recuperação completa ou no modelo de recuperação bulk-logged, faz o backup do log de transações do banco de dados para criar um backup de log (BACKUP LOG).

Veja a sintaxe do comando BACKUP (Transact-SQL):

```
SQL

--Backing Up a Whole Database
BACKUP DATABASE { database_name | @database_name_var }
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL
        | <general_WITH_options> [ ,...n ] } ]
[;]

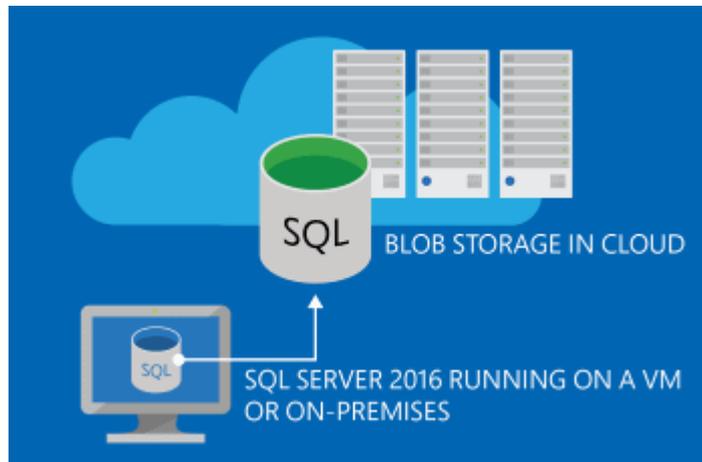
--Backing Up Specific Files or Filegroups
BACKUP DATABASE { database_name | @database_name_var }
  <file_or_filegroup> [ ,...n ]
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL | <general_WITH_options> [ ,...n ] } ]
[;]

--Creating a Partial Backup
BACKUP DATABASE { database_name | @database_name_var }
  READ_WRITE_FILEGROUPS [ , <read_only_filegroup> [ ,...n ] ]
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL | <general_WITH_options> [ ,...n ] } ]
[;]

--Backing Up the Transaction Log (full and bulk-logged recovery models)
BACKUP LOG
  { database_name | @database_name_var }
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { <general_WITH_options> | \<log-specific_optionspec> } [ ,...n ] ]
[;]
```

Além do armazenamento local para guardar os backups, o SQL Server também oferece suporte ao backup e à restauração no serviço de armazenamento de Blob do Windows Azure.





O SQL Server dá suporte ao armazenamento de backups no serviço de armazenamento de Blobs do Microsoft Azure, das seguintes maneiras:

Gerenciamento dos backups para o Microsoft Azure: usando os mesmos métodos usados para fazer backup em DISCO e FITA, agora você pode fazer backup para o armazenamento do Microsoft Azure especificando a URL como o destino do backup. Você pode usar esse recurso para fazer backup ou configurar manualmente sua própria estratégia de backup, como faria em um armazenamento local ou em outras opções fora do local. Esse recurso também é conhecido como Backup do SQL Server para URL e oferece desempenho e funcionalidade com o uso de blobs de bloco, assinaturas de acesso compartilhado e distribuição.

Backups de instantâneo de arquivo de arquivos de banco de dados no Armazenamento de Blobs do Azure: Com o uso de instantâneos do Azure, os Backups de Instantâneo de Arquivo do SQL Server fornecem backups e restaurações instantâneos de arquivos de banco de dados armazenados por meio do serviço de armazenamento de Blobs do Azure.

Deixando o SQL Server gerenciar os backups para o Microsoft Azure: configure o SQL Server para gerenciar a estratégia de backup e agendar backups para um único banco de dados ou vários bancos de dados, ou defina valores padrão no nível da instância. Esse recurso é conhecido como **Backup gerenciado do SQL Server** no Microsoft Azure.

Com o uso de blobs de bloco no SQL Server, você pode distribuir o conjunto de backup para dar suporte a arquivos de backup com tamanho de até 12,8 TB.

O arquivo de backup que, agora, é armazenado no serviço de armazenamento de Blobs do Microsoft Azure está diretamente disponível para um SQL Server local ou outro SQL Server em execução em uma Máquina Virtual do Microsoft Azure, sem que seja necessário anexar/desanexar o banco de dados ou baixar e anexar o VHD.

5. Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

No Sql Server, um becape

- a) de arquivo agrega um ou mais arquivos ou grupos de arquivos de banco de dados.
- b) parcial agrega dados que ainda não foram afetados por COMMIT, tanto de dados quanto de transações.



- c) completo de banco de dados agrega todos os dados de todos os bancos de dados no momento em que o becape é concluído, com exceção dos logs de transação.
- d) diferencial agrega apenas logs de transações, incluindo somente transações feitas desde seu último becape de log até a transação mais recente.
- e) de logs de transações inclui todos os registros de log de forma cumulativa, independentemente de ter havido becape de log anterior ou becape completo.

Comentário: SQL Server faz backup de um banco de dados completo do SQL Server para criar um backup de banco de dados ou um ou mais arquivos ou grupos de arquivos do banco de dados para criar um backup de arquivo (BACKUP DATABASE).

Além disso, no modelo de recuperação completa ou no modelo de recuperação bulk-logged, faz o backup do log de transações do banco de dados para criar um backup de log (BACKUP LOG).

Assim, temos o gabarito na letra a).

Gabarito: A

6. Inspetor de Controle Externo (TCE-RN)/Tecnologia da Informação/2015

No que se refere a tecnologia e arquitetura de banco de dados, julgue o próximo item.

No MSSQL Server 2014, o recurso AlwaysOn é uma solução de alta disponibilidade e de recuperação de desastres que fornece uma alternativa, em nível corporativo, para o espelhamento de bancos de dados, a partir do gerenciamento de réplicas de bancos de dados.

Certo

Errado

Comentário: Habilitar os Grupos de Disponibilidade AlwaysOn é pré-requisito para uma instância do SQL Server usar grupos de disponibilidade como uma solução de recuperação de desastres de alta disponibilidade.

Os recursos secundários ativos do Grupos de disponibilidade AlwaysOn incluem suporte para execução de operações de backup em réplicas secundárias. As operações de backup podem colocar tensão significativa na E/S e na CPU (com compactação de backup). O descarregamento de backups em uma réplica secundária sincronizada ou em sincronização permite usar os recursos na instância do servidor que hospeda a réplica primária para suas cargas de trabalho de camada-1.

Logo, o gabarito da questão é certo.

Gabarito: C



CONSTRUINDO UM BANCO DE DADOS

Nesta parte da aula vamos construir o quebra cabeça das principais partes utilizadas para compor um banco de dados. Vamos passar pelos conceitos de Schema, tipos de dados, restrições de integridade até chegarmos à criação de uma tabela propriamente dita. Em seguida avançaremos para o conceito de índice em banco de dados SQL Server.

O CONCEITO DE SCHEMA

Enquanto um banco de dados é o recipiente primário de todos os objetos, esquemas oferecem outro nível de agrupamento e organização dentro de um banco de dados. Utilizando um esquema, um usuário pode agrupar objetos de escopo ou propriedade semelhante. Por padrão, o proprietário banco de dados (dbo) do esquema é automaticamente criado dentro de um banco de dados. Qualquer objeto que é criado é adicionado a este esquema.

Veja abaixo um exemplo do comando de criação de esquema:

```
CREATE SCHEMA Concursos;  
GO
```

TIPOS DE DADOS DO SQL SERVER

SQL Server contém quatro categorias de tipos de dados: Numéricos, Strings, Data e tempo e outros. Cada uma das quatro categorias contém subcategorias. Todas as colunas numa tabela, declarações de variáveis e parâmetros devem ter um tipo de dados correspondente. Um tipo de dados simplesmente especifica o tipo de dados pode alocar o objeto (coluna, variável, parâmetro, e assim por diante). A integridade de dados depende de uma escolha apropriada dos tipos de dados; portanto, você não deve depender ou confiar em um aplicativo para impor tipo de dados. Vamos agora conhecer cada um deles.

TIPOS DE DADOS NUMÉRICOS

Os tipos de dados numéricos têm duas subcategorias: exatos e aproximados. Tipos de dados exatos cabem dentro de um intervalo finito de números. A tabela a seguir listas e define cada tipo de dados numéricos exatos.



| Data Type | Range | Storage |
|-------------------|---|---------|
| <i>bigint</i> | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 | 8 bytes |
| <i>int</i> | -2,147,483,648 to 2,147,483,647 | 4 bytes |
| <i>smallint</i> | -32,768 to 32,767 | 2 bytes |
| <i>tinyint</i> | 0 to 255 | 1 byte |
| <i>money</i> | -922,337,203,685,477.5808 to 922,337,203,685,477.5807 | 8 bytes |
| <i>smallmoney</i> | -214,748.3648 to 214,748.3647 | 4 bytes |

Se você precisar de uma coluna em uma tabela que armazena apenas valores entre 1 e 10, você deve usar um **tinyint**. Além dos tipos de dados na tabela, a categoria de números exatos inclui mais de dois tipos de dados: **decimal** e **numeric**. Eles são um pouco diferentes dos outros na medida em que permitem o uso de casas decimais, que são baseadas em dois valores: precisão e escala.

Essencialmente, **decimal(p,s)** e **numeric(p,s)** são muito semelhantes na forma como eles armazenam dados. A **precisão** é o número total de dígitos que podem ser armazenadas em ambos os lados da casa decimal. Este valor só pode ser entre 1 e 38. A **escala** corresponde ao número de dígitos que pode ser armazenado à direita da casa decimal e só é especificado quando a precisão é fornecida. Este valor irá estar compreendido entre zero e a precisão especificada. Portanto, se você quiser armazenar um número de quatro dígitos, com apenas dois dígitos à direita da casa decimal, você usaria decimal (4,2). Vejam que esse assunto já foi cobrado em prova:

Tipos de dados numéricos aproximados são utilizados para dados numéricos de ponto flutuante. O ponto flutuante é uma aproximação, portanto, nem todos os valores do tipo de dados podem ser representados exatamente. Quando tratamos de SQL Server temos dois tipos de dados de ponto flutuante o **real** e o **float**. O sinônimo ISO para **real** é **float (24)**. Observem que o SQL Server **não** apresenta o tipo double. Todos os números de pontos flutuantes criados são representados por meio do tipo float. Se você quiser aumentar a precisão basta mudar o valor da variável passada como parâmetros.

A sintaxe para criação do tipo é float [(n)]. Onde n é o número de bits que são usados para armazenar a mantissa do número flutuante em notação científica e, portanto, determina o tamanho de precisão e o espaço de armazenamento. Se n for especificado, ele deve ser um valor entre 1 e 53. O valor padrão de n é de 53. Veja abaixo uma tabela com o relacionamento entre o valor de n, a precisão e o espaço de armazenamento:

| n value | Precision | Storage size |
|---------|-----------|--------------|
| 1-24 | 7 digits | 4 bytes |
| 25-53 | 15 digits | 8 bytes |



TIPOS DE DADOS DE STRING

O tipo de dados sequência de caracteres ou string contém três subcategorias: `character`, `Unicode` e `binary`. Cada uma contém três tipos de dados específicos. Os tipos de dados são semelhantes em que cada subcategoria. Um tipo de dados com comprimento fixo, um tipo de dados de comprimento variável, e outro que foi descontinuado. Um parâmetro `n` define o comprimento da cadeia que pode ser armazenado. Para os tipos de dados de comprimento variável, o valor de `n` indica o tamanho máximo a ser armazenado que pode ser de até 2 GB. Vamos falar um pouco sobre cada um deles:

A subcategoria `character` vai armazenar dados não-Unicode. Os três tipos são os seguintes:

`char (n)` indica uma sequência de caracteres de comprimento fixo, esse tipo de dados possui um comprimento de cadeia que varia entre 1 e 8.000.

`varchar(n)` representa um tipo de dados de sequência de caracteres com comprimento variável que pode armazenar até 2 GB de dados.

`text` - tipo de dados *deprecated*. Substitua-o por um `varchar (max)`.

A sequência de caracteres da subcategoria `Unicode` irá armazenar dados `Unicode` e não-`Unicode`. Também tem três tipos de dados:

`nchar(n)` apresenta uma sequência de caracteres de comprimento fixo, esse tipo de dados possui um comprimento de cadeia variando entre 1 e 4.000.

`nvarchar(n)` tipo de dados sequência de caracteres de comprimento variável que pode armazenar até 2 GB de dados.

`ntext` - tipo de dados *deprecated*. Substitua-o por `nvarchar(max)`.

A subcategoria cadeia binária irá armazenar dados binários. Seus três tipos são os seguintes:

`binary(n)` tipo de dados de comprimento fixo para armazenamento de valores binários com um comprimento de cadeia entre 1 e 8.000.

`varbinary(n)` tipo de dados binários de comprimento variável com tamanho máximo de cadeia de até 2 GB.

`imagem` - tipo de dados descontinuado. Substitua por `varbinary (max)`.

Como boa prática de projeto de banco de dados, você deve usar tipos de dados com comprimento fixo (`char`, `nchar`, `binary`) em todas as variáveis do tipo *string*, quando os valores que serão armazenados têm um tamanho consistente. Quando os valores não são consistentes, você deve usar os tipos de dados de comprimento variável (`varchar`, `nvarchar`, `varbinary`).

TIPOS DE DADO PARA DATA E HORA

Os tipos de dados para armazenamento de **data** e **hora** são amplamente utilizados em bancos de dados SQL Server. Eles oferecem a conveniência de armazenar a data e hora de várias maneiras. Há seis tipos de dados nesta categoria.



time(n) Este tipo de dados armazena a hora do dia sem a consciência de fuso horário com base em um relógio de 24 horas. Time, aceita um argumento, que é a precisão fracionária de segundos. Você só pode fornecer valores entre 0 e 7. Conforme o número aumenta, o mesmo acontece com a precisão fracionária. Se você especificar time(2), você pode armazenar um valor semelhante ao 11:51:04:24. Mudando de 2 a 3 a precisão aumenta para três casas, semelhante à 11:51:04:245.

date Este tipo de dados armazena um valor de data entre 01-01-01 e 31-12-9999.

smalldatetime Este tipo de dados armazena um valor de data e hora. O valor da data é entre 1/1/1900 e 06/06/2079. A precisão de tempo é reduzida para segundos. Um valor possível para smalldatetime é 4/1/2015 21:15:04 em portanto, pode ser armazenados usando este tipo de dados. Vejam que esse assunto já foi questão de prova

datetime Este tipo de dados é semelhante ao smalldatetime, mas oferece um intervalo de datas maior e uma maior precisão com relação ao tempo. Ele oferece a mesma gama data que o parâmetro de data, 01-01-01 a 12-31-9999, e que tem um valor mais preciso de tempo, com três casas decimais nos milésimos de segundo. Um valor de 2012/04/01 11:15:04:888 podem ser armazenados usando este tipo de dados.

datetime2(n) Este tipo de dados é semelhante ao datetime, mas oferece uma flexibilidade maior no range de tempo. Ao contrário do datetime, você pode controlar o valor da precisão fracionária dos segundos. Você só pode fornecer valores entre 0 e 7. Se você especificar datetime2(2), você pode armazenar um valor semelhante ao 2012/04/01 11:51:04:24. Mudando de 2 a 3 aumenta a precisão para três números, semelhante à 2012/04/01 11:51:04:249.

datetimeoffset Este tipo de dados inclui todos os recursos de datetime2, e também tem consciência fuso horário (*time zone*). Isso o torna único entre os tipos de dados de data e hora. Usando este tipo de dados, você pode armazenar o deslocamento do fuso horário juntamente com a data e hora. Um valor de 2012/04/01 03:10:24-06:00 podem ser armazenados usando este tipo de dados.

1. Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2013

Um novo tipo de dado introduzido no MS-SQL SERVER 2008 possibilita o controle de fusos horários e deve ser preenchido com um dos padrões estabelecidos pela ISO 8601. Este novo campo é o

- a) jetleg.
- b) isodate.
- c) datefuse.
- d) datetimeoffset.
- e) datetime2.

Comentário: O tipo de dado datetimeoffset (Transact-SQL) define a data combinada com uma hora de um dia que possui reconhecimento de fuso horário e é baseada em um relógio de 24 horas.



Formatos de literal de cadeia de caracteres padrão (usados para cliente de nível inferior): YYYY-MM-DD hh:mm:ss[.nnnnnnn] [{+|-}hh:mm]

Dessa forma, o gabarito está na letra d).

Gabarito: D

EXTENSÕES PARA DADOS ESPECIAIS (OUTROS)

Além dos tipos de dados detalhados nas seções anteriores, o SQL Server inclui vários outros tipos de dados. Abaixo apresentamos uma lista que relaciona cada tipo de dados adicional com uma breve descrição.

cursor Uma cópia temporária de dados que vai ser utilizada para os processos recursivos ou iterativos. De todos os tipos de dados, este é o único que não pode ser incluído como parte de uma tabela.

rowversion(timestamp) Este tipo de dados gera automaticamente um valor de 8 bytes semelhante ao 0x0000000000000001. *rowversion* substitui o tipo de dados *timestamp*, que foi *deprecated*. Este tipo de dados é tipicamente utilizado para detectar alterações nos dados.

hierarchyid Este é um tipo de dados de posicionamento. Ela representa uma posição em uma hierarquia. *hierarchyid* é usado para organizar dados, como uma lista de materiais e organogramas.

sql_variant Este é o coringa dos tipos de dados. *sql_variant* pode assumir a identidade de praticamente qualquer tipo de dados na lista de tipos de dados do SQL Server. Antes de realizar quaisquer operações sobre ele, você deve convertê-lo para o respectivo tipo de dados a ser utilizado. Por exemplo, se você quiser executar uma adição, você deve transformar este tipo de dados em um int ou algum outro tipo de dados numérico que suporte essa operação.

XML Você pode armazenar dados XML usando este tipo de dados.

geoespacial SQL Server suporta dois tipos de dados geoespaciais: GEOGRAPHY e GEOMETRY. GEOGRAPHY representa dados na um sistema de coordenadas sobre a Terra. GEOMETRY é um tipo de dados plano no qual você pode armazenar pontos, linhas e outras figuras geométricas.

filestream Este tipo de dados permite armazenar dados não estruturados comuns, como documentos e imagens. SQL Server foi acoplado com o sistema de arquivos NTFS, permitindo o armazenamento de variáveis do tipo varbinary(max) no sistema de arquivos.



(Ministério da Economia – Desenvolvimento de Sistemas - 2020) Acerca de sistemas gerenciadores de banco de dados, julgue o item subsequente.



O SQL Server tem um tipo de dado denominado `sql_variant`, que armazena dados de imagem e texto de tamanho grande, maior que 10 KMbytes.

Comentários: O tipo de dados `sql_variant` permite que uma coluna da tabela ou uma variável mantenha valores de **qualquer tipo de dados** com um comprimento máximo de 8000 bytes mais 16 bytes que contêm as informações do tipo de dados.

A lista a seguir mostra os tipos de dados que **não** permitidos em uma coluna ou variável `sql_variant`:

- `varchar(max)`
- `varbinary(max)`
- `nvarchar(max)`
- `xml`
- `text`
- `ntext`
- **image**
- `rowversion (timestamp)`
- `geography`
- `hierarchyid`
- `geometry`
- `datetimeoffset`
- User-defined types

Gabarito Errado.

2. Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2013

Espaciais são dados que identificam formas e locais geográficos. Entre eles podem estar pontos de referência, estradas e até mesmo o local de uma empresa. O MS-SQL Server 2008 fornece dois tipos de dados para trabalhar com este recurso:

- a) geoposition e altitude.
- b) latitude e longitude.
- c) gps e glonass.
- d) geography e geometry.
- e) xcord e ycord.

Comentário: O SQL Server oferece o recurso de GeometryCollection . Uma GeometryCollection é uma coleção de zero ou mais instâncias de geometry ou de geography . Uma GeometryCollection pode estar vazia.

Os dados espaciais representam informações sobre o local físico e a forma de objetos geométricos. Esses objetos podem ser locais de pontos ou objetos mais complexos como países, estradas ou lagos.

SQL Server dá suporte a dois tipos de dados espaciais: geometria e geografia .

O tipo geometria representa dados em um sistema de coordenadas euclidiano (plano).



O tipo geografia representa dados em um sistema de coordenadas esféricas.
Assim, o gabarito da questão está na letra d).

Gabarito: D

DATA DEFINITION LANGUAGE (DDL)

Após nossa rápida conversa sobre tipos de dados vamos agora começar a construção dos nossos objetos no SQL Server. A Data Definition Language (DDL) é um vocabulário usado para definir estruturas de dados em SQL Server. Usamos estas instruções para criar, alterar ou remover as estruturas de dados em uma instância do SQL Server. Começaremos pela criação de uma tabela.

CRIANDO UMA TABELA (CREATE)

A primeira parte do comando de criação de tabelas é semelhante ao comando utilizado para criação de todos os demais objetos de SQL Server:

```
CREATE <object type> <object name>
```

Nesta seção queremos tratar apenas de tabelas, então seremos mais específicos:

```
CREATE TABLE Concursos
```

Se lembrarmos do comando CREATE DATABASE tratado anteriormente, para sua definição precisaríamos apenas desta primeira linha. Os demais parâmetros seriam atribuídos de acordo com os valores default definidos no SQL Server para um modelo de DATABASE. Com tabelas, porém, não existe nenhum modelo, então você deve se preocupar em definir mais algumas informações como colunas, tipos de dados e operadores especiais.

Vejam abaixo uma figura com a sintaxe do comando CREATE TABLE.

```
CREATE TABLE [database_name].[owner].table_name  
(<column name> <data type>  
[[DEFAULT <constant expression>]  
|[IDENTITY [(seed, increment) [NOT FOR REPLICATION]]]]  
[ROWGUIDCOL]  
[COLLATE <collation name>]  
[NULL|NOT NULL]  
[<column constraints>  
|[column_name AS computed_column_expression]  
|[<table_constraint>]  
[,...n]  
)  
[ON {<filegroup>|DEFAULT}]  
[TEXTIMAGE_ON {<filegroup>|DEFAULT}]
```



A definição acima está incompleta do ponto de vista de possibilidades sintáticas do comando. Caso você queira conhecer a definição completa pode verificar [aqui](#). Contudo as descrições acima nos ajudam a ter um entendimento do comando. Vamos observar cada uma das partes começando pela segunda linha.

Há uma preocupação em definir nomes para as colunas e tabelas. As regras para criação dos nomes desses objetos são as mesmas aplicadas a banco de dados. Depois da definição do nome, você deve fornecer um tipo de dados que será atribuído a coluna. Já falamos sobre os tipos de dados do SQL Server.

Caso você queira definir um valor padrão para uma coluna você pode utilizar o comando DEFAULT seguido pelo valor desejado. Esse valor será atribuído para a coluna caso não seja passado como parâmetro quando for inserida uma nova linha nesta tabela.

Outra possibilidade muito importante para o design de banco de dados é a definição da cláusula IDENTITY. Quando você define uma coluna com IDENTITY o SQL Server imediatamente designa um número de sequência para cada nova linha inserida. O valor inicial da contagem é conhecido com *seed*, esse valor é incrementado por outro parâmetro denominado *increment*. O valor default para seed e increment é 1.

Uma coluna IDENTITY deve ser do tipo numérico, e, na prática, é quase sempre implementado com um inteiro ou bigint. O uso é bastante simples: você simplesmente inclui a palavra-chave IDENTITY logo após o tipo de dados atribuído para a coluna. A opção IDENTITY não pode ser usada em conjunto com a restrição DEFAULT. Faz sentido se você pensar sobre isso - como pode haver um padrão constante, se você está incrementando a variável?

Colunas do tipo de IDENTITY são geralmente usadas como chave primária, mas isso não é obrigatório.

O parâmetro NOT FOR REPLICATION determina que um novo valor de IDENTITY será criado no caso de replicação da tabela em outro banco de dados. Caso ele não seja incluído a tabela replicada terá os mesmos valores para a coluna.

Outro parâmetro usado para replicação é o ROWGUIDCOL ele tem o mesmo propósito de uma coluna IDENTITY. É usado para identificar exclusivamente cada linha em uma tabela. A diferença é que o sistema vai ter certeza que o valor utilizado é verdadeiramente único. Em vez de usar uma contagem numérica, o SQL Server usa o conhecido *unique identifier* (na verdade GUID significa Globally Unique Identifier).

Embora um valor do identificador seja normalmente exclusivo através do tempo, não é único no espaço. Portanto, você pode ter duas cópias de sua tabela em execução, e pode ter em ambas os valores de identificadores idênticos atribuídos para linhas diferentes. Isso provoca grandes problemas quando você tenta juntar linhas de ambas as tabelas em uma única tabela replicada. O identificador único, conhecido também como GUID, é único em todo espaço e no tempo.

O COLLATE funciona praticamente da mesma forma que comentamos no comando CREATE DATABASE. A principal diferença é em termos de escopo. Aqui, você define o parâmetro no nível da coluna em vez do nível de banco de dados.



Quando você define uma coluna como NULL ou NOT NULL você restringe ou não o uso do valor nulo, presente em todos os tipos de dados, para a coluna especificada. O padrão, caso não seja especificado, é NOT NULL.

Sobre as restrições de integridade, podemos definir as chaves primária e estrangeira, restrições de CHECK, UNIQUE e DEFAULT. Falaremos um pouco sobre elas quando comentarmos sobre o comando ALTER TABLE.

A outra opção do comando CREATE é atribuir a uma coluna um valor calculado. Neste caso o valor não é armazenado, ele é calculado no momento em que alguma consulta sobre a tabela é feita. Vamos dar uma olhada na sintaxe do comando:

```
<column name> AS <computed column expression>
```

O nome da coluna fornece um nome ser associado ao valor. Este é simplesmente um alias que você vai usar para se referir ao valor que é calculado, com base na expressão que segue a palavra-chave AS.

Em seguida, vem a expressão da coluna computada. A expressão pode ser qualquer expressão que usa valores literais ou valores de colunas a partir da mesma tabela. Portanto, podemos definir uma coluna que calcula o preço multiplicado pela quantidade como:

```
PrecoTotal AS Preco * Quantidade
```

A próxima definição refere-se as restrições de tabela. São as mesmas utilizadas para definições das colunas. A diferença é que agora podemos criar restrições que englobam mais de uma coluna da tabela.

A cláusula ON em uma definição de tabela é uma forma de dizer especificamente sobre a qual grupo de arquivos (filegroup) você deseja que a tabela esteja localizada. Você pode colocar uma determinada tabela em um dispositivo físico específico ou, como você vai querer fazer na maioria dos casos, basta não utilizar a cláusula ON, e ela é colocada no filegroup padrão.

Quando você usa a cláusula TEXTIMAGE_ON, você se move apenas as informações BLOB para o grupo de arquivos separado - o resto da tabela permanece tanto no filegroup padrão ou com o grupo de arquivos escolhidos na cláusula ON.

Vamos agora mostrar um exemplo prático da criação de uma tabela:

```
CREATE TABLE Employees
(
EmployeeID int          IDENTITY      NOT NULL,
FirstName  varchar(25)   NOT NULL,
MiddleInitial char(1)    NULL,
LastName  varchar(25)   NOT NULL,
Title     varchar(25)   NOT NULL,
SSN       varchar(11)   NOT NULL,
Salary    money          NOT NULL,
PriorSalary money       NOT NULL,
LastRaise AS Salary - PriorSalary,
```



ESCLARECENDO

```
HireDate    date                NOT NULL,  
TerminationDate date          NULL,  
ManagerEmpID int          NOT NULL,  
Department    varchar(25)          NOT NULL  
)
```

ALTERANDO UMA TABELA (ALTER)

Ok, então agora você tem um banco de dados e um par de tabelas. Seria ótimo que as tabelas sempre permanecessem inalteradas. Às vezes, na verdade, com muito mais frequência do que você gostaria você vai ter pedidos para alterar uma tabela em vez de recriá-la. Da mesma forma, você pode precisar alterar o tamanho, locais de arquivo, ou alguma outra característica de seu banco de dados. É aí que entra a instrução ALTER.

Você pode antes de fazer a alteração de uma tabela usar o `sp_help` para verificar as definições atuais para a tabela. `EXEC sp_help <nome_da_tabela>`.

Uma necessidade muito, muito mais comum é a situação em que você precisa mudar a estrutura de uma das suas tabelas. Isto pode variar de tarefas simples como adicionar uma nova coluna a questões mais complexas, como a alteração de um tipo de dados. Vamos começar dando uma olhada na sintaxe utilizada alterar uma tabela:

Vejam que basicamente temos a possibilidade de alterar uma coluna já existente ou adicionar uma nova coluna. Suponha que você queira adicionar uma nova coluna a tabela. Isso pode ser feito por meio do seguinte comando:

```
ALTER TABLE Employees  
ADD  
PreviousEmployer varchar(30) NULL
```

É possível adicionar mais de uma coluna em apenas um comando de ALTER TABLE. Você pode observar que todas as colunas são adicionadas no final da lista de colunas. Não há nenhuma maneira de adicionar uma coluna em um local específico no SQL Server. Se você quiser mover uma coluna para o meio da tabela, você precisa criar uma tabela (com um nome diferente), copiar os dados para a nova tabela, apagar a tabela existente e, em seguida, renomear a nova tabela para o nome antigo.

```
ALTER TABLE table_name  
{[ALTER COLUMN column_name  
{ [schema of new data type].new_data_type  
[(precision [, scale)]] max |  
<xml schema collection>  
[COLLATE collation_name]  
[NULL|NOT NULL]  
|[{ADD|DROP} ROWGUIDCOL] | PERSISTED}]  
|ADD  
<column name> <data_type>
```



```

[[DEFAULT <constant_expression>]
|[IDENTITY [(<seed>, <increment>) [NOT FOR REPLICATION]]]]
[ROWGUIDCOL]
[COLLATE <collation_name>]
[NULL|NOT NULL]
[<column_constraints>]
|[<column_name> AS <computed_column_expression>]
|ADD
[CONSTRAINT <constraint_name>]
{{{PRIMARY KEY|UNIQUE}
[CLUSTERED|NONCLUSTERED]
{(<column_name>[,...n ])}
[WITH FILLFACTOR = <fillfactor>]
[ON {<filegroup> | DEFAULT}]
]
|FOREIGN KEY
[(<column_name>[,...n])]
REFERENCES <referenced_table> [(<referenced_column>[,...n])]
[ON DELETE {CASCADE|NO ACTION}]
[ON UPDATE {CASCADE|NO ACTION}]
[NOT FOR REPLICATION]
|DEFAULT <constant_expression>
[FOR <column_name>]
|CHECK [NOT FOR REPLICATION]
(<search_conditions>)
[,...n][,...n]
|[WITH CHECK|WITH NOCHECK]
| { ENABLE | DISABLE } TRIGGER
{ ALL | <trigger name> [,...n] }
|DROP
[[CONSTRAINT] <constraint_name>
|COLUMN <column_name>[,...n]
|{CHECK|NOCHECK} CONSTRAINT
{ALL|<constraint_name>[,...n]}
|{ENABLE|DISABLE} TRIGGER
{ALL|<trigger_name>[,...n]}
| SWITCH [ PARTITION <source partition number expression> ]
TO [ schema_name. ] target_table
[ PARTITION <target partition number expression> ]
}
}

```



Vamos agora tratar das alterações de tabela que definem restrições de integridade. Antes, porém, é importante lembrar que as restrições são classificadas em três tipos: restrições de entidade, restrições de domínio e restrições de integridade referencial.

As restrições de domínio lidam com uma ou mais colunas. O importante aqui é garantir que uma determinada coluna ou conjunto de colunas obedeçam a critérios específicos. Quando você inserir ou atualizar uma linha, a restrição é aplicada. Você vai ver este tipo de restrição ao lidar com restrições de CHECK, RULE e DEFAULT.

Restrições entidade são todas sobre a comparação de linhas. Esta forma de restrição não se preocupa com o que os dados da coluna. Ela está interessada em uma determinada linha, e como comparar a linha com outras linhas na mesma tabela. Restrições entidade são exemplificadas por uma restrição que exige que cada linha tenha um valor único para uma coluna ou combinação de colunas. Você vai obter este tipo de restrição ao lidar com chave primária e restrições do tipo UNIQUE.

Restrições de integridade referencial são criadas quando um valor em uma coluna deve corresponder ao valor de outra coluna quer na mesma tabela ou, muito mais tipicamente, em uma tabela diferente.

Vamos agora observar a criação de algumas restrições utilizando o comando ALTER TABLE. Começando pela restrição de chave primária, observem que o comando solicita que você defina um nome para a constraint, neste caso denominada PK_Employees.

```
USE Accounting
```

```
ALTER TABLE Employees
```

```
ADD CONSTRAINT PK_Employees
```

```
PRIMARY KEY (EmployeeID);
```

As chaves estrangeiras são tanto um método de garantir a integridade dos dados e uma manifestação das relações entre tabelas. Quando você adiciona uma chave estrangeira a uma tabela, você está criando uma dependência entre a tabela para a qual você define a chave estrangeira e a tabela onde as referências de chave estrangeira estão. Depois de adicionar uma chave estrangeira, qualquer registro que você inserir na tabela de referência deve ter um registro correspondente na(s) coluna(s) referenciada(s) da outra tabela, ou o valor da coluna de chave estrangeira deve ser definido como NULL. Isso pode ser um pouco confuso, então vamos dar uma olhada em um exemplo.

```
ALTER TABLE Orders
```

```
ADD CONSTRAINT FK_EmployeeCreatesOrder
```

Restrições UNIQUE é, essencialmente, o irmão mais novo de chaves primárias que eles exigem um valor único para toda a coluna (ou combinação de colunas) na tabela. Muitas vezes você vai ouvir



restrições UNIQUE referidas chaves como alternativas. As principais diferenças são que eles não são considerados para ser o identificador original de um registro em uma tabela (embora você possa efetivamente usá-lo dessa forma), assim você pode ter mais de uma restrição UNIQUE (lembre-se que você só pode ter uma chave primária por tabela).

Depois de estabelecer uma restrição UNIQUE, todos os valores nas colunas devem ser exclusivos. Se você tentar atualizar ou inserir uma linha com um valor que já existe em uma coluna com uma restrição UNIQUE, o SQL Server irá gerar um erro e rejeitar o registro. Veja um exemplo da adição de uma restrição UNIQUE:

```
ALTER TABLE Employees  
ADD CONSTRAINT AK_EmployeeSSN
```

A última restrição que falaremos é a de CHECK. Uma coisa interessante sobre as restrições CHECK é que elas não estão restritas a uma determinada coluna. Elas podem ser relacionadas com uma coluna, mas também podem ser essencialmente da tabela. Nestes casos pode-se verificar uma coluna contra a outra, desde que todas as colunas estejam dentro de uma única tabela e os valores comparados estão na mesma linha a ser atualizada ou inserida. Eles também podem verificar se qualquer combinação de valores da coluna possui um critério determinado. Vejamos um exemplo abaixo.

```
ALTER TABLE Customers  
ADD CONSTRAINT CN_CustomerDateInSystem  
CHECK
```

Dica: O SQL Server não cria a restrição, a menos que os dados existentes satisfaçam os critérios de restrição. Para contornar esta situação é preciso corrigir os dados existentes ou você pode fazer uso da opção **WITH NOCHECK** na instrução ALTER.

3. Técnico Judiciário (TRT 23ª Região)/Apoio Especializado/Tecnologia da Informação/2016

Atenção: Para responder à questão, considere as informações abaixo.

Um Técnico está participando da modelagem de um banco de dados utilizando o Modelo Entidade-Relacionamento – MER e se deparou, dentre outras, com a entidade Processo, que contém os seguintes atributos:

NumeroProcesso – inteiro (PK)

DigitoProcesso – inteiro (PK)

AnoProcesso – inteiro (PK)



NumeroOABAdvogadoProcesso – cadeia de caracteres

NomeAdvogadoProcesso – cadeia de caracteres

NumeroOrgaoJudiciarioProcesso – inteiro (FK)

NumeroTribunal – inteiro (FK)

NumeroUnidadeOrigemProcesso – inteiro (FK)

Após criar a tabela Processo no Sistema Gerenciador de Banco de Dados SQL Server, para definir uma restrição que especifica que o campo AnoProcesso só poderá receber números inteiros maiores do que 2014, o Técnico deve utilizar a instrução

- a) ADD CONSTRAINT Processo CHECK (AnoProcesso>2014);
- b) ALTER TABLE Processo ADD CHECK (AnoProcesso>2014);
- c) ADD CONSTRAINT (AnoProcesso>2014) FROM Processo;
- d) CREATE CONSTRAINT Chk_Processo FROM Processo CHECK (AnoProcesso>2014);
- e) ALTER TABLE Processo ADD CONSTRAINT (AnoProcesso>2014);

Comentário: A instrução ALTER TABLE column_constraint especifica as propriedades de uma restrição PRIMARY KEY, FOREIGN KEY, UNIQUE ou CHECK que faz parte da definição de uma nova coluna adicionada a uma tabela usando ALTER TABE. Veja a sintaxe:

```
ALTER TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
{
    ALTER COLUMN column_name
    {
        [ type_schema_name. ] type_name
        [ (
            {
                precision [ , scale ]
                | max
                | xml_schema_collection
            }
        ) ]
        [ COLLATE collation_name ]
        [ NULL | NOT NULL ] [ SPARSE ]
        | { ADD | DROP }
        { ROWGUIDCOL | PERSISTED | NOT FOR REPLICATION | SPARSE | HIDDEN }
        | { ADD | DROP } MASKED [ WITH ( FUNCTION = ' mask_function ' ) ]
    }
    [ WITH ( ONLINE = ON | OFF ) ]
    | [ WITH { CHECK | NOCHECK } ]

    | ADD
    {
        <column_definition>
        | <computed_column_definition>
        | <table_constraint>
        | <column_set_definition>
    } [ ,...n ]
```



```
[ CONSTRAINT constraint_name ]  
{  
  [ NULL | NOT NULL ]  
  { PRIMARY KEY | UNIQUE }  
    [ CLUSTERED | NONCLUSTERED ]  
    [ WITH FILLFACTOR = fillfactor ]  
    [ WITH ( index_option [, ...n ] ) ]  
    [ ON { partition_scheme_name (partition_column_name)  
        | filegroup | "default" } ]  
  | [ FOREIGN KEY ]  
    REFERENCES [ schema_name . ] referenced_table_name  
      [ ( ref_column ) ]  
    [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  
    [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  
    [ NOT FOR REPLICATION ]  
  | CHECK [ NOT FOR REPLICATION ] ( logical_expression )  
}
```

O enunciado da questão exige o uso do argumento CHECK, o qual é uma restrição que impõe a integridade de domínio limitando os possíveis valores que podem ser inseridos em uma ou mais colunas.

logical_expression: É uma expressão lógica usada em uma restrição CHECK e retorna TRUE ou FALSE. logical_expression usada com restrições CHECK não pode fazer referência a outra tabela, mas pode fazer referência a outras colunas na mesma tabela para a mesma linha. A expressão não pode referenciar um tipo de dados de alias.

Assim, temos o gabarito na letra b):

```
ALTER TABLE Processo ADD CHECK (AnoProcesso>2014)
```

logical_expression: AnoProcesso>2014

Gabarito: B

O COMANDO DROP

Executar um DROP é o mesmo que excluir objeto(s) que você faz referência na declaração DROP. É muito rápido e fácil, e a sintaxe é exatamente a mesma para todos os principais objetos do SQL Server (tabelas, visões, store procedures, triggers, e assim por diante). Veja o padrão a seguir:

```
DROP <object type> <object name> [, ...n]
```

Na verdade, isso é quase tão simples quanto as instruções DROP do SQL padrão. Você pode remover duas tabelas ao mesmo tempo, se você quiser, basta usar o comando como se segue:

```
USE conta
```

```
DROP TABLE Clientes, Colaboradores
```



DATA MANIPULATION LANGUAGE (DML)

A DML (linguagem de manipulação de dados) afeta as informações armazenadas no banco de dados. Use estas instruções para inserir, atualizar e alterar as linhas no banco de dados.

O COMANDO DELETE

Remove uma ou mais linhas de uma tabela ou exibição no SQL Server. Sintaxe:

```
[ WITH <common_table_expression> [ ,...n ] ]
DELETE
  [ TOP ( expression ) [ PERCENT ] ]
  [ FROM ]
  { { table_alias
    | <object>
    | rowset_function_limited
    [ WITH ( table_hint_limited [ ...n ] ) ] }
    | @table_variable
  }
  [ <OUTPUT Clause> ]
  [ FROM table_source [ ,...n ] ]
  [ WHERE { <search_condition>
    | { [ CURRENT OF
        { { [ GLOBAL ] cursor_name }
          | cursor_variable_name
        }
      }
    ]
  }
  ]
  [ OPTION ( <Query Hint> [ ,...n ] ) ]
[; ]

<object> ::=
{
  [ server_name.database_name.schema_name.
  | database_name. [ schema_name ] .
  | schema_name.
  ]
  table_or_view_name
}
```

WITH <common_table_expression>



Especifica o conjunto de resultados nomeados temporário, também conhecido como expressão de tabela comum, definido dentro do escopo da instrução DELETE. O conjunto de resultados é derivado de uma instrução SELECT.

Também podem ser usadas expressões de tabela comuns com as instruções SELECT, INSERT, UPDATE e CREATE VIEW.

WITH (<table_hint_limited> [... n])

Especifica uma ou mais dicas de tabela permitidas para uma tabela de destino. A palavra-chave WITH e parênteses são necessários. NOLOCK e READUNCOMMITTED não são permitidos.

FROM table_source

Especifica uma cláusula FROM adicional. Essa extensão Transact-SQL para DELETE permite especificar dados de <table_source> e excluir as linhas correspondentes da tabela na primeira cláusula FROM.

WHERE

Especifica as condições usadas para limitar o número de linhas que são excluídas. Se uma cláusula WHERE não for fornecida, DELETE removerá todas as linhas da tabela.

Para excluir todas as linhas em uma tabela, use TRUNCATE TABLE. TRUNCATE TABLE é mais rápido que DELETE e usa menos recursos do sistema e do log de transações.

O COMANDO INSERT

Adiciona uma ou mais linhas a uma tabela ou exibição no SQL Server. Sintaxe:



```
[ WITH <common_table_expression> [ ,...n ] ]  
INSERT  
{  
    [ TOP ( expression ) [ PERCENT ] ]  
    [ INTO ]  
    { <object> | rowset_function_limited  
      [ WITH ( <Table_Hint_Limited> [ ...n ] ) ]  
    }  
    {  
        [ ( column_list ) ]  
        [ <OUTPUT Clause> ]  
        { VALUES ( { DEFAULT | NULL | expression } [ ,...n ] ) [ ,...n ]  
          | derived_table  
          | execute_statement  
          | <dml_table_source>  
          | DEFAULT VALUES  
        }  
    }  
}  
[;]  
  
<object> ::=  
{  
    [ server_name . database_name . schema_name .  
      | database_name .[ schema_name ] .  
      | schema_name .  
    ]  
    table_or_view_name  
}
```

```
<dml_table_source> ::=  
    SELECT <select_list>  
    FROM ( <dml_statement_with_output_clause> )  
        [AS] table_alias [ ( column_alias [ ,...n ] ) ]  
    [ WHERE <search_condition> ]  
        [ OPTION ( <query_hint> [ ,...n ] ) ]
```

WITH <common_table_expression>

Especifica o conjunto de resultados nomeado temporário, também conhecido como expressão de tabela comum, definido dentro do escopo da instrução INSERT. O conjunto de resultados é derivado de uma instrução SELECT.

WITH (<table_hint_limited> [... n])

Especifica uma ou mais dicas de tabela permitidas para uma tabela de destino. A palavra-chave WITH e parênteses são necessários.

(column_list)



É uma lista de uma ou mais colunas onde os dados devem ser inseridos. `column_list` deve ser colocada entre parênteses e separada por vírgulas.

VALUES

Apresenta a(s) lista(s) de valores de dados a serem inseridos. Deve haver um valor de dados para cada coluna em `column_list`, se especificado, ou na tabela. A lista de valores deve ser colocada entre parênteses.

O COMANDO UPDATE

Altera dados existentes em uma tabela ou exibição no SQL Server 2017. Sintaxe:

```
[ WITH <common_table_expression> [...n] ]
UPDATE
  [ TOP ( expression ) [ PERCENT ] ]
  { { table_alias | <object> | rowset_function_limited
    [ WITH ( <Table_Hint_Limited> [ ...n ] ) ]
  }
  | @table_variable
}
SET
  { column_name = { expression | DEFAULT | NULL }
  | { udt_column_name. { { property_name = expression
                        | field_name = expression }
                        | method_name ( argument [ ,...n ] )
                      }
  }
  | column_name { .WRITE ( expression , @Offset , @Length ) }
  | @variable = expression
  | @variable = column = expression
  | column_name { += | -= | *= | /= | %= | &= | ^= | |= } expression
  | @variable { += | -= | *= | /= | %= | &= | ^= | |= } expression
  | @variable = column { += | -= | *= | /= | %= | &= | ^= | |= } expression
} [ ,...n ]

[ <OUTPUT Clause> ]
[ FROM { <table_source> } [ ,...n ] ]
[ WHERE { <search_condition>
  | { [ CURRENT OF
      { { [ GLOBAL ] cursor_name }
        | cursor_variable_name
      }
    }
  ]
}
```



```
]
[ OPTION ( <query_hint> [ ,...n ] ) ]
[ ; ]

<object> ::=
{
  [ server_name . database_name . schema_name .
  | database_name .[ schema_name ] .
  | schema_name .
  ]
  table_or_view_name}
```

WITH <common_table_expression>

Especifica a exibição ou o conjunto de resultados nomeado temporário, também conhecido como CTE (expressão de tabela comum), definido dentro do escopo da instrução UPDATE. O conjunto de resultados da CTE é derivado de uma consulta simples e é referido pela instrução UPDATE.

Expressões de tabela comuns também podem ser usadas com as instruções SELECT, INSERT, DELETE e CREATE VIEW.

DEFAULT

Especifica que o valor padrão definido para a coluna deve substituir o valor existente na coluna. Isso também poderá ser usado para alterar a coluna para NULL se ela não tiver nenhum padrão e estiver definida para permitir valores nulos.

FROM <table_source>

Especifica que uma tabela, exibição ou origem de tabela derivada é usada para fornecer os critérios da operação de atualização.

Uma exibição com um gatilho INSTEAD OF UPDATE não pode ser um destino de UPDATE com uma cláusula FROM.

WHERE

Especifica os critérios que limitam as linhas que são atualizadas.

<search_condition>

Especifica o critério a ser atendido para as linhas a serem atualizadas. O critério de pesquisa também pode ser o critério no qual uma junção é baseada. Não há nenhum limite para o número de predicados que podem ser incluídos em um critério de pesquisa.

O COMANDO TRUNCATE TABLE

Remove todas as linhas de uma tabela ou partições especificadas de uma tabela sem registrar as exclusões de linha individual. TRUNCATE TABLE é semelhante à instrução DELETE sem nenhuma



cláusula WHERE; entretanto, TRUNCATE TABLE é mais rápida e utiliza menos recursos de sistema e log de transações. Sintaxe:

```
TRUNCATE TABLE
  [ { database_name . [ schema_name ] . | schema_name . } ]
  table_name
  [ WITH ( PARTITIONS ( { <partition_number_expression> | <range> }
    [ , ...n ] ) ) ]
  [ ; ]

<range> ::=
  <partition_number_expression> TO <partition_number_expression>
```

Para truncar uma tabela particionada, a tabela e os índices deverão estar alinhados (particionados na mesma função de partição).

PRINCIPAIS STORED PROCEDURES

No SQL Server 2017, muitas atividades administrativas e informativas podem ser executadas com os procedimentos armazenados do sistema. Stored Procedure ou Procedimento Armazenado é um conjunto de comandos em SQL que podem ser executados de uma só vez, como em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.

Um Stored Procedure pode reduzir o tráfego na rede, melhorar a performance de um banco de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento, etc. O SQL fornece procedimentos armazenados ligados a diversas categorias, como Procedimentos armazenados do catálogo, Procedimentos armazenados de coletor de dados e Procedimentos armazenados de consultas distribuídas.

Veja um exemplo de um stored procedure que executa uma consulta utilizando um filtro por descrição, em uma tabela específica de um banco de dados:

```
USE BancoDados
GO
CREATE PROCEDURE Busca --- Declarando o nome da procedure
@CampoBusca VARCHAR (20) --- Declarando variável (note que utilizamos o @ antes do nome
da variável)
AS
SELECT Codigo, Descrição --- Consulta
FROM NomeTabela
WHERE Descricao = @CampoBusca --- Utilizando variável como filtro para a consulta
```

Como o objeto da aula é o mecanismo de banco de dados, vamos ver alguns dos procedimentos armazenados desta categoria, usados para manutenção geral do Mecanismo de Banco de Dados do SQL Server.



sp_add_log_file_recover_suspect_db (Transact-SQL)

Adiciona um arquivo de log a um grupo de arquivos quando a recuperação não pode ser concluída em um banco de dados devido a espaço insuficiente de log (erro 9002). Depois que o arquivo é adicionado, sp_add_log_file_recover_suspect_db desativa a configuração suspeita e conclui a recuperação do banco de dados. Os parâmetros são os mesmos de ALTER DATABASE database_name ADD LOG FILE. Sintaxe:

```
sp_add_log_file_recover_suspect_db [ @dbName= ] 'database' ,  
  [ @name = ] 'logical_file_name' ,  
  [ @filename= ] 'os_file_name' ,  
  [ @size = ] 'size' ,  
  [ @maxsize = ] 'max_size' ,  
  [ @filegrowth = ] 'growth_increment'
```

sp_configure (Transact-SQL)

Exibe ou altera parâmetros de configuração global para o servidor atual. Sintaxe:

```
sp_configure [ [ @configname = ] 'option_name'  
  [ , [ @configvalue = ] 'value' ] ]
```

[**@configname=**] 'option_name'

É o nome de uma opção de configuração. option_name é varchar(35), com um padrão de NULL.

sp_helpdb (Transact-SQL)

Relata informações sobre um banco de dados especificado ou todos os bancos de dados. Sintaxe:

```
sp_helpdb [ [ @dbname= ] 'name' ]
```

[**@dbname=**] 'name'

É o nome do banco de dados cujas informações são reportadas.

PRINCIPAIS FUNÇÕES

SQL Server fornece os seguintes grupos de funções do sistema:

- Funções de grupos de disponibilidade Always On
- Funções do Change Data Capture
- Funções de controle de alterações
- Funções do coletor de dados



- Funções FileStream e FileTable
- Funções de Backup gerenciadas
- sys.fn_get_sql
- sys.fn_MSxe_read_event_stream
- sys.fn_stmt_sql_handle_from_sql_stmt
- sys.fn_validate_plan_guide
- sys.fn_xe_file_target_read_file
- sys.fn_backup_file_snapshots
- Funções de pesquisa semântica de texto completo
- Funções de metadados do sistema
- Funções de segurança do sistema
- Funções de rastreamento do sistema

Para melhor entendimento desta parte da aula, veremos as funções disponíveis relacionadas a grupos de disponibilidade Always On:

sys.fn_hadr_is_primary_replica

Usado para determinar se a réplica atual for a réplica primária. Sintaxe:

```
sys.fn_hadr_is_primary_replica ( 'dbname' )
```

'dbname': É o nome do banco de dados. DBName é do tipo sysname.

sys.fn_hadr_backup_is_preferred_replica

Usado para determinar se a réplica atual for a réplica de backup preferencial. Sintaxe:

```
sys.fn_hadr_backup_is_preferred_replica ( 'dbname' )
```

'dbname': É o nome do banco de dados do qual é feito o backup. DBName é do tipo sysname.

sys.fn_hadr_distributed_ag_replica

Usado para mapear uma réplica em um grupo de disponibilidade distribuído para o grupo de disponibilidade local. Sintaxe:

```
sys.fn_hadr_distributed_ag_replica( lag_Id, replica_id )
```

'lag_Id': É o identificador do grupo de disponibilidade distribuído. lag_Id é do tipo uniqueidentifier.

'replica_id': É o identificador de uma réplica no grupo de disponibilidade distribuído. replica_id é do tipo uniqueidentifier.



sys.fn_hadr_distributed_ag_database_replica

Usado para mapear um banco de dados em um grupo de disponibilidade distribuído para o banco de dados no grupo de disponibilidade local. Sintaxe:

```
sys.fn_hadr_distributed_ag_database_replica( lag_Id, database_id )
```

'lag_Id': É o identificador do grupo de disponibilidade distribuído. lag_Id é do tipo uniqueidentifier.

'database_id': É o identificador do banco de dados em um grupo de disponibilidade distribuído. database_id é do tipo uniqueidentifier.

TRANSACT SQL (T-SQL)

O Transact-SQL é uma parte central no uso do SQL Server. Todos os aplicativos que se comunicam com uma instância do SQL Server o fazem enviando instruções Transact-SQL ao servidor, independentemente da interface do usuário do aplicativo. A seguir, apresentamos uma lista dos tipos de aplicativos que podem gerar Transact-SQL:

- Aplicativos de produtividade para escritórios em geral.
- Aplicativos que usam uma interface gráfica do usuário para permitir que os usuários selecionem as tabelas e colunas que querem visualizar dados.
- Aplicativos que usam sentenças de linguagem comum para determinar quais dados um usuário deseja consultar.
- Linha de aplicativos empresariais que armazenam os dados em bancos de dados SQL Server. Esses aplicativos podem incluir tanto aplicativos escritos por fornecedores como aplicativos escritos internamente.
- Scripts Transact-SQL que são executados com utilitários, como [sqlcmd](#).
- Aplicativos criados usando sistemas de desenvolvimento como Microsoft Visual C++, Microsoft Visual Basic ou Microsoft Visual J++ que usam APIs de banco de dados como ADO, OLE DB e ODBC.
- Páginas da Web que extraem dados de bancos de dados SQL Server.
- Sistemas de banco de dados distribuídos dos quais são replicados dados do SQL Server para vários bancos de dados ou são executadas consultas distribuídas.
- Data warehouses nos quais dados são extraídos das bases de dados de processamento de transações online (OLTP) e são resumidos para análise de suporte à decisão.

4. Analista Judiciário (TRF 3ª Região)/Apoio Especializado/Informática - Banco de Dados/2014

No SQL Server 2012, os gatilhos DDL são disparados em resposta a diversos eventos DDL. Esses eventos correspondem principalmente as instruções Transact-SQL que começam com algumas palavras-chave como



- a) INSERT e DELETE.
- b) UPDATE e INSERT.
- c) GRANT e DENY.
- d) SELECT e UNION.
- e) INNER JOIN e ALTER TABLE.

Comentário: Os gatilhos DDL são disparados em resposta a diversos eventos DDL (linguagem de definição de dados). Esses eventos correspondem principalmente a instruções Transact-SQL que começam com as palavras-chave CREATE, ALTER, DROP, GRANT, DENY, REVOKE ou UPDATE STATISTICS. Determinados procedimentos armazenados do sistema que executam operações do tipo DDL também podem disparar gatilhos DDL.

Use gatilhos DDL quando quiser fazer o seguinte:

- Evitar determinadas alterações em seu esquema de banco de dados.
- Ocorrer algo no banco de dados em resposta a uma alteração em seu esquema de banco de dados.
- Registrar alterações ou eventos no esquema de banco de dados.

Assim, temos o gabarito na letra c).

Gabarito: C

SEGURANÇA NO SQL SERVER

O SQL Server fornece algumas instruções específicas para prover segurança. Vamos falar sobre cada uma delas abaixo:

ADD SIGNATURE - Esta função foi modificada no SQL Server 2014. É possível adicionar uma assinatura digital a um procedimento armazenado, função, assembly ou gatilho. Ou ainda uma *countersignature* aos mesmos objetos. Uma countersignature é basicamente uma confirmação, por meio de outra assinatura, de um documento que já está assinado. Podemos compara isso com um cartório que faz autenticação de assinatura garantindo que, quem assina o documento, é quem de fato escreveu.

GRANT - Concede permissões. A sintaxe geral é GRANT <alguma permissão> ON <algum objeto> para <algum usuário, login, ou grupo>. Veja a sintaxe do comando abaixo:

```
Simplified syntax for GRANT
GRANT { ALL [ PRIVILEGES ] }
      | permission [ ( column [ ,...n ] ) ] [ ,...n ]
      [ ON [ class :: ] securable ] TO principal [ ,...n ]
      [ WITH GRANT OPTION ] [ AS principal ]
```



O argumento ALL vai garantir todas as permissões que podem variar dependendo do objeto sobre o qual deseja conceder a permissão.

Se o objeto é um banco de dados, ALL significa DATABASE BACKUP, BACKUP LOG, CREATE DATABASE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE RULE, CREATE TABLE e CREATE VIEW. Se o objeto é uma função escalar, ALL refere-se a EXECUTE e REFERENCES. Se o objeto é uma função com valor de tabela, ALL significa DELETE, INSERT, REFERENCES, SELECT e UPDATE. Se o objeto é um procedimento armazenado, ALL permite EXECUTE. Se o objeto for uma tabela, ALL indica os privilégios de DELETE, INSERT, REFERENCES, SELECT e UPDATE.

A chave mestra do banco de dados (MASTER KEY) é uma chave simétrica usada para proteger as chaves privadas de certificados e as chaves assimétricas que estão presentes na base de dados. Quando ele é criado, a chave mestra é criptografada usando o algoritmo AES_256 e uma senha fornecida pelo usuário. Quando um banco de dados é restaurado faz-se necessário utilizar o comando **OPEN MASTER KEY** para decifrar a chave mestra do banco de dados. A sintaxe do comando segue abaixo:

```
OPEN MASTER KEY DECRYPTION BY PASSWORD = 'password'
```

CLOSE MASTER KEY - Esta declaração inverte a operação realizada pelo OPEN MASTER KEY. Ela só consegue executar quando a chave mestra de banco de dados foi aberta na sessão atual usando a instrução OPEN MASTER KEY.

DENY - Nega uma permissão a uma entidade de segurança. Impede a entidade de segurança de herdar a permissão através das suas associações de grupo ou de função.

EXECUTE AS - Por padrão, uma sessão é iniciada quando um usuário faz logon e é encerrada quando ele faz logoff. Todas as operações durante uma sessão estão sujeitas a verificações de permissão para aquele usuário. Quando uma instrução EXECUTE AS é executada, o contexto de execução da sessão é alternado para o logon ou nome de usuário especificado. Depois da alternância de contexto, as permissões são verificadas no logon e nos tokens de segurança do novo usuário, em vez da pessoa que chama a instrução EXECUTE AS. Em essência, o usuário ou a conta de logon são utilizados durante a execução da sessão ou do módulo, ou a alternância de contexto é explicitamente revertida por meio do comando **REVERT**.

REVOKE – Ele revoga as permissões concedidas pelo GRANT ou uma instrução DENY para um determinado usuário ou grupo. Veja a sintaxe do comando abaixo.



```
Simplified syntax for REVOKE
REVOKE [ GRANT OPTION FOR ]
  {
    [ ALL [ PRIVILEGES ] ]
    |
    permission [ ( column [ ,...n ] ) ] [ ,...n ]
  }
[ ON [ class :: ] securable ]
{ TO | FROM } principal [ ,...n ]
[ CASCADE ] [ AS principal ]
```



SQL SERVER NA NUVEM

Sumário

| | |
|---|----|
| SQL Server na Nuvem | 2 |
| Recursos de Banco de Dados em Servidores MS-SQL SERVER..... | 2 |
| Plataforma de Dados em Nuvem | 4 |
| Soluções Híbridas do MSSQL Server | 8 |
| Particionamento de Tabela | 11 |
| Estratégias de Migração Offline ou Online | 18 |
| Sincronização de Dados SQL para Azure..... | 22 |



SQL SERVER NA NUVEM

Vamos começar esse bloco da nossa aula revisando algumas das funcionalidades do SQL Server. O Microsoft SQL Server é uma poderosa plataforma de gerenciamento de banco de dados relacional que oferece uma ampla gama de recursos para garantir a eficiência, segurança e escalabilidade dos dados. Entre os recursos disponíveis no MS-SQL SERVER, destacam-se:

1. **Recursos de Armazenamento Avançados:** o SQL Server oferece opções avançadas de armazenamento, como tabelas particionadas, compressão de dados e índices colunares, que permitem otimizar o desempenho e o uso do espaço em disco.
2. **Recursos de Segurança:** o SQL Server oferece recursos robustos de segurança, incluindo autenticação integrada do Windows, criptografia de dados, controle de acesso baseado em funções e auditoria avançada para proteger os dados contra acessos não autorizados.
3. **Recursos de Backup e Recuperação:** o SQL Server possui ferramentas e recursos para garantir a integridade dos dados, como backups automáticos, logs de transações, pontos de verificação e opções flexíveis de restauração para minimizar a perda de dados em caso de falhas.
4. **Recursos de Monitoramento e Otimização de Desempenho:** o SQL Server oferece ferramentas para monitorar o desempenho do banco de dados, identificar gargalos, otimizar consultas e índices, e manter o banco de dados funcionando de forma eficiente.
5. **Recursos de Gerenciamento de Dados em Nuvem:** com integração com serviços de nuvem como o Microsoft Azure, o SQL Server oferece recursos para facilitar a migração de dados para a nuvem, bem como opções para executar bancos de dados diretamente na nuvem.

Esses são apenas alguns dos recursos disponíveis no MS-SQL SERVER para o gerenciamento eficaz de banco de dados. Com sua variedade de funcionalidades avançadas, o SQL Server é uma escolha popular para organizações que buscam um sistema de gerenciamento de banco de dados confiável e robusto. Nesta aula, tentaremos mostrar um pouco dos recursos de nuvem do SQL Server e como eles podem ajudar na criação de bancos de dados mais eficientes.

Recursos de Banco de Dados em Servidores MS-SQL SERVER

A versão mais recente do SQL Server é o SQL Server 2022. Ele foi lançado em novembro de 2021 e traz consigo várias melhorias e novos recursos, como melhorias de desempenho, segurança aprimorada e recursos avançados de gerenciamento de dados. Essa versão oferece maior escalabilidade, confiabilidade e eficiência para atender às demandas de bancos de dados modernos e complexos.

No SQL Server 2022, a Microsoft uma gama abrangente de recursos robustos para atender às necessidades de armazenamento, processamento e análise de dados. Abaixo estão os principais recursos destacados no SQL Server 2022:

1. **Mecanismo de Banco de Dados do SQL Server:** O Mecanismo de Banco de Dados é o coração do SQL Server, responsável pelo armazenamento, processamento e proteção de dados. Ele oferece recursos avançados como replicação, pesquisa de texto completo e ferramentas para gerenciar dados XML e



relacionais. Além disso, integra análise de banco de dados e suporta PolyBase para acesso a fontes de dados heterogêneas.

O PolyBase é um recurso poderoso disponível no SQL Server que permite consultar e combinar dados de diversas fontes de dados diferentes, sejam elas estruturadas ou não estruturadas, sem a necessidade de movimentar ou transformar os dados. É uma tecnologia de virtualização de dados que simplifica o acesso e a análise de informações armazenadas em diferentes sistemas, como bancos de dados relacionais, arquivos Hadoop, ou mesmo dados armazenados em serviços de nuvem.

Com o PolyBase, os usuários podem usar a linguagem SQL padrão para consultar dados de fontes heterogêneas, agregando as informações de diversas fontes em uma única consulta. Isso permite uma integração de dados mais eficiente e simplificada, facilitando análises abrangentes e a extração de insights valiosos a partir de conjuntos de dados diversos.

2. **Serviços de Analysis:** Os Serviços de Analysis permitem a criação e gerenciamento de aplicativos OLAP e mineração de dados. Eles fornecem recursos essenciais para análise de dados e geração de insights valiosos. O Analysis Services proporciona a capacidade essencial de criar cubos OLAP, que são estruturas multidimensionais que facilitam a análise interativa e a exploração de dados com várias dimensões. Esses cubos permitem visualizar informações de maneira hierárquica e analisar dados sob diferentes perspectivas.
3. **Reporting Services:** O Reporting Services inclui componentes para criação, gerenciamento e implantação de uma variedade de relatórios, como tabulares, de matriz, gráficos e de forma livre. Ele é uma plataforma extensível e essencial para desenvolver aplicativos de relatório personalizados.
4. **Integration Services:** O Integration Services oferece ferramentas gráficas e programáveis para mover, copiar e transformar dados. Ele também incorpora o componente Data Quality Services (DQS) para garantir a qualidade dos dados utilizados.
5. **Master Data Services (MDS):** O Master Data Services é a solução do SQL Server para gerenciar dados mestres, permitindo a configuração e gerenciamento de vários domínios de dados. Ele oferece recursos como hierarquias, segurança granular, transações e controle de versão dos dados.
6. **Serviços de Machine Learning (No Banco de Dados):** Os Serviços de Machine Learning no Banco de Dados suportam soluções escalonáveis de aprendizado de máquina usando fontes de dados empresariais. O SQL Server 2022 oferece suporte tanto para a linguagem R quanto para o Python, possibilitando análises avançadas.
7. **Virtualização de Dados com o PolyBase:** O PolyBase permite a consulta de diferentes tipos de dados em várias fontes de dados do SQL Server, facilitando a integração e análise de dados de diversas origens.
8. **Serviços Conectados do Azure:** Os Serviços Conectados do Azure estendem as capacidades do SQL Server 2022 para integração com serviços e recursos do Microsoft Azure, como Azure Synapse Link, políticas de acesso do Microsoft Purview e pagamento conforme o uso, proporcionando uma integração contínua com a infraestrutura em nuvem.



O Azure Synapse Link é um recurso poderoso que permite conectar e integrar de forma transparente dados entre o Azure Synapse Analytics e os bancos de dados do SQL Server. Esse recurso é uma extensão do Azure Synapse Analytics que permite acessar e analisar em tempo real dados operacionais diretamente no SQL Server, sem a necessidade de mover ou copiar os dados.

O Microsoft Purview é uma plataforma de governança de dados avançada e automatizada fornecida pela Microsoft. Ele foi projetado para ajudar as organizações a descobrir, mapear e gerenciar seus ativos de dados em toda a empresa, independentemente de onde esses dados estejam armazenados.

Esses recursos abrangentes do SQL Server 2022 fornecem uma plataforma sólida e completa para armazenar, analisar e extrair insights valiosos dos dados em uma organização. Nas próximas sessão, vamos explorar o Azure, plataforma de nuvem da Microsoft no contexto de análise de dados.

Plataforma de Dados em Nuvem

SQL Server 2022 cloud-connected capabilities



A plataforma de nuvem da Microsoft, conhecida como Microsoft Azure, é uma infraestrutura escalável em nuvem que oferece uma ampla gama de serviços e recursos para empresas de todos os portes. O Azure é projetado para ajudar as organizações a aumentar a produtividade, inovar rapidamente e gerenciar seus aplicativos e dados de forma eficiente na nuvem.

Com data centers distribuídos globalmente (inclusive no Brasil), o Microsoft Azure proporciona uma infraestrutura confiável e segura para hospedar aplicativos, armazenar dados, realizar análises avançadas e implantar serviços em escala. A plataforma oferece flexibilidade para escolher entre uma variedade de serviços, como computação, armazenamento, banco de dados, inteligência artificial, Internet das Coisas (IoT) e muito mais, permitindo que as empresas construam, gerenciem e dimensionem soluções complexas.



Além disso, o Azure oferece recursos avançados de segurança, conformidade e governança para proteger os dados e garantir a privacidade dos usuários. Com ferramentas de monitoramento e gerenciamento integradas, as organizações podem otimizar o desempenho de seus aplicativos e assegurar a continuidade dos negócios.

Para criar um banco de dados no Azure, você pode seguir os passos básicos abaixo:

1. **Acessar o Portal do Azure:** Acesse o Portal do Azure em <https://portal.azure.com/> e faça login na sua conta. Após a criação da conta, você poderá acessar o Portal do Azure e começar a usar os serviços e recursos disponíveis gratuitamente ou por meio de uma assinatura paga, dependendo das suas necessidades. Se você realmente for usar a ferramenta, lembre-se de monitorar seus custos e utilizar os créditos gratuitos da assinatura de maneira responsável.
2. **Criar um Grupo de Recursos (Resource Group):** Antes de criar o banco de dados, é recomendável criar um Grupo de Recursos para organizar todos os recursos relacionados. Você pode criar um Grupo de Recursos na seção "Grupos de Recursos" no portal. Um Grupo de Recursos no Microsoft Azure é um contêiner que mantém todos os recursos relacionados a um aplicativo ou projeto em um único local lógico. Ele funciona como uma unidade de gerenciamento para organizar, administrar e controlar todos os recursos associados em um ambiente de nuvem.
3. **Criar um Banco de Dados SQL:** No portal do Azure, clique em "Criar um recurso" e pesquise por "Banco de Dados SQL do Azure". Siga as instruções fornecidas para configurar o banco de dados, como escolher a subscrição, o Grupo de Recursos, o tipo de implantação (único banco de dados, pool elástico etc.), e outras configurações necessárias.

Escolha um nome exclusivo para o seu banco de dados, selecione uma região onde ele será hospedado e defina as opções de nível de desempenho e escalabilidade de acordo com suas necessidades. Configure as opções de segurança, como nome de usuário e senha para acesso ao banco de dados.

4. **Revisar e Criar o Banco de Dados:** Revise todas as configurações e opções escolhidas, e então clique em "Criar" para provisionar o banco de dados no Azure.
5. **Acessar e Gerenciar o Banco de Dados:** Após a conclusão do processo de criação, você poderá acessar e gerenciar seu banco de dados por meio do Portal do Azure ou por outras ferramentas de gerenciamento de bancos de dados, como o Azure Data Studio ou o SQL Server Management Studio.

Esses são os passos básicos para criar um banco de dados no Azure. Certifique-se de revisar os preços e custos associados ao uso do banco de dados, pois o Azure pode cobrar pelos recursos consumidos. Fique atento às configurações de segurança e permissões para garantir a proteção dos seus dados e informações sensíveis.

Configuração de Banco de Dados em Ambiente de Nuvem



Ao configurar um banco de dados em um ambiente de nuvem Azure, existem alguns aspectos específicos a serem considerados. Aqui estão alguns dos principais pontos a serem observados:

1. **Escolha do serviço de banco de dados:** A Azure oferece uma variedade de serviços de banco de dados, como Azure SQL Database, Cosmos DB, MySQL, PostgreSQL, entre outros. É importante escolher o serviço mais adequado às necessidades do seu aplicativo.
2. **Configuração de rede:** Definir corretamente as regras de firewall e rede para garantir que somente aplicativos autorizados possam acessar o banco de dados na nuvem.
3. **Segurança e conformidade:** Implementar medidas de segurança como criptografia de dados, controle de acesso baseado em função e conformidade com padrões de segurança, como o GDPR, dependendo dos requisitos da sua organização.
4. **Dimensionamento automático:** Aproveitar recursos como a escala automática para ajustar dinamicamente a capacidade do banco de dados com base na carga de trabalho, garantindo desempenho e eficiência.

O dimensionamento automático no Azure para banco de dados é uma funcionalidade que permite ajustar dinamicamente a capacidade de um banco de dados com base na carga de trabalho e nas necessidades de desempenho. Existem algumas opções de dimensionamento automático que podem ser utilizadas dependendo do serviço de banco de dados escolhido no Azure, como o Azure SQL Database.

No Azure SQL Database, por exemplo, o dimensionamento automático pode ser configurado de duas maneiras principais:

- **Autoescalabilidade horizontal (Elastic Pool):** Com essa opção, é possível agrupar vários bancos de dados relacionados em um pool elástico, permitindo que os recursos de computação e armazenamento sejam compartilhados entre eles. O Azure ajusta automaticamente a quantidade de recursos alocados com base na carga de trabalho agregada dos bancos de dados no pool.
- **Autoescalabilidade vertical (Banco de Dados Único):** Para um banco de dados único, é possível configurar o dimensionamento automático vertical com base na carga de trabalho. Isso significa que o Azure pode ajustar automaticamente a quantidade de CPU e armazenamento alocados para o banco de dados para atender à demanda.

O dimensionamento automático no Azure ajuda a otimizar os custos, garantir o desempenho e a disponibilidade do banco de dados, eliminando a necessidade de ajustes manuais frequentes. Com a configuração correta do dimensionamento automático, os recursos do banco de dados podem ser ajustados de forma dinâmica, sem intervenção manual, proporcionando uma experiência mais eficiente e escalável.



5. **Monitoramento e gerenciamento:** Utilizar ferramentas de monitoramento da Azure para acompanhar o desempenho do banco de dados, identificar problemas e otimizar consultas conforme necessário.
6. **Backup e recuperação:** Configurar políticas de backup regulares e testar procedimentos de recuperação de desastres para garantir a disponibilidade contínua dos dados.

Esses são alguns dos principais aspectos a considerar ao configurar um banco de dados em um ambiente de nuvem Azure. Cada aplicação e cenário terá requisitos específicos que devem ser levados em conta durante a configuração.

Para finalizar, é importante ter em mente que a utilização de ambientes de nuvem para bancos de dados oferece diversas vantagens significativas, tais como:

1. **Elasticidade e escalabilidade:** Os ambientes de nuvem permitem escalar vertical ou horizontalmente os recursos do banco de dados de acordo com a demanda, garantindo alta disponibilidade e desempenho otimizado.
2. **Redução de custos:** Em vez de investir em infraestrutura de hardware e manutenção local, as empresas podem utilizar modelos de pagamento por uso na nuvem, reduzindo custos operacionais.
3. **Flexibilidade:** Os ambientes de nuvem oferecem flexibilidade para adaptar a configuração do banco de dados às necessidades específicas do negócio, permitindo mudanças rápidas e fáceis conforme necessário.
4. **Segurança:** Muitos provedores de nuvem possuem certificações de segurança e oferecem recursos avançados de proteção de dados, criptografia e conformidade regulatória para manter os dados seguros.
5. **Backup e recuperação:** A maioria dos ambientes de nuvem oferece soluções automatizadas de backup e recuperação de dados, garantindo a integridade e disponibilidade das informações.
6. **Colaboração e acesso remoto:** Os ambientes de nuvem permitem acesso remoto aos dados, facilitando o trabalho colaborativo entre equipes distribuídas geograficamente.
7. **Atualizações e manutenção:** O provedor de nuvem é responsável por manter a infraestrutura e aplicar atualizações de software, liberando a equipe de TI para focar em outras tarefas críticas.

Essas são apenas algumas das vantagens da utilização de ambientes de nuvem para bancos de dados. Ao migrar os dados para a nuvem, as empresas podem obter maior agilidade, eficiência e segurança em suas operações de banco de dados.



Soluções Híbridas do MSSQL Server

As soluções híbridas do Microsoft SQL Server oferecem uma abordagem flexível que combina implantações locais (on-premises) com recursos de nuvem para atender às necessidades das empresas que buscam uma transição gradual para a nuvem ou que precisam de uma combinação de ambientes. Aqui estão alguns pontos-chave sobre as soluções híbridas do MSSQL Server:

1. **Azure SQL Database:** O Azure SQL Database é a oferta de banco de dados relacional como serviço (DBaaS) na nuvem da Microsoft. Ele permite hospedar instâncias do SQL Server na nuvem, oferecendo escalabilidade, alta disponibilidade e integração com outros serviços do Azure.
2. **Azure SQL Managed Instance:** Esta é uma opção do Azure SQL Database que fornece uma instância totalmente gerenciada do SQL Server na nuvem, mantendo a compatibilidade com recursos locais como Linked Servers, Agent Jobs e Ferramentas de Gerenciamento.
3. **SQL Server Stretch Database:** Essa funcionalidade permite estender parte dos seus dados do SQL Server para a nuvem, mantendo-os acessíveis para consulta, mas minimizando a utilização de recursos locais. É útil para armazenar dados históricos ou menos acessados.

O SQL Server Stretch Database é uma funcionalidade do Microsoft SQL Server que permite estender parte dos dados de um banco de dados local para o Microsoft Azure. Essa funcionalidade foi projetada para ajudar as empresas a gerenciar grandes volumes de dados, mantendo o custo e o desempenho sob controle. Aqui estão alguns detalhes sobre o SQL Server Stretch Database:

Funcionamento: Com o Stretch Database, os dados menos acessados ou históricos são identificados e movidos de forma transparente para o Azure, enquanto uma referência aos dados é mantida no banco de dados local. Quando consultas são feitas envolvendo esses dados estendidos, o SQL Server automaticamente busca os dados relevantes no Azure e os combina com os dados locais antes de retornar o resultado da consulta.

Principais Benefícios:

- **Otimização de Custo:** Ao estender dados menos acessados para o Azure, as empresas podem economizar custos em armazenamento local, mantendo apenas os dados mais ativos localmente.
- **Desempenho Otimizado:** O SQL Server gerencia automaticamente a movimentação de dados, garantindo que consultas envolvendo dados estendidos sejam eficientes e rápidas.
- **Gerenciamento Simplificado:** A funcionalidade é integrada ao SQL Server Management Studio, facilitando o monitoramento e a administração dos dados estendidos.
- **Segurança:** Todos os dados estendidos são criptografados para garantir a segurança e a conformidade dos dados.



Cenários de Uso: Empresas com grande volume de dados históricos podem usar o Stretch Database para manter o acesso a esses dados sem comprometer o desempenho do banco de dados local. Aplicações que exigem retenção prolongada de dados, como em conformidade com regulamentações, podem se beneficiar da capacidade de armazenar dados históricos de forma eficaz.

O SQL Server Stretch Database é uma ferramenta eficaz para otimizar o armazenamento e o desempenho de bancos de dados, permitindo que as empresas gerenciem grandes volumes de dados de forma eficiente e econômica, mantendo a flexibilidade e a escalabilidade necessárias para suas operações.

4. **Always On Availability Groups com Réplicas no Azure:** Com a configuração de Always On Availability Groups, você pode configurar réplicas de um banco de dados local no Azure para fins de recuperação de desastres, alta disponibilidade ou balanceamento de carga.

O Always On Availability Groups com Réplicas no Azure é uma funcionalidade do Microsoft SQL Server que permite configurar réplicas de alta disponibilidade e recuperação de desastres em ambientes híbridos, mantendo cópias dos dados críticos sincronizados entre instâncias locais e na nuvem (Azure). Aqui estão detalhes sobre essa configuração:

Funcionamento: Com o Always On Availability Groups, é possível criar um grupo de disponibilidade que consiste em várias instâncias do SQL Server (primária e replicações secundárias). As réplicas secundárias podem ser configuradas no Azure para fornecer uma estratégia de alta disponibilidade e recuperação de desastres. Os dados são sincronizados entre as réplicas de forma contínua, garantindo a consistência dos dados e a capacidade de failover automático em caso de falha da instância primária.

Benefícios:

- **Alta Disponibilidade:** O Always On Availability Groups oferece uma solução robusta de alta disponibilidade, garantindo que os dados estejam sempre disponíveis, mesmo em caso de falha de uma instância.
- **Recuperação de Desastres:** Com as réplicas no Azure, as empresas podem proteger seus dados críticos e garantir a recuperação rápida em caso de desastres que afetem os sistemas locais.
- **Escalabilidade:** As réplicas secundárias no Azure permitem escalar a capacidade de processamento e armazenamento conforme necessário, garantindo o desempenho ideal do banco de dados.
- **Baixo Tempo de Recuperação (RTO):** O failover automático para réplicas no Azure reduz o tempo de inatividade e minimiza o impacto nas operações em caso de falha.

Cenários de Uso: Empresas com requisitos críticos de alta disponibilidade podem se beneficiar do Always On Availability Groups com réplicas no Azure para garantir a continuidade dos negócios.



Empresas que exigem uma estratégia de recuperação de desastres eficaz podem usar essa configuração para proteger seus dados contra perdas e garantir a retomada rápida das operações.

Em resumo, o Always On Availability Groups com réplicas no Azure oferece uma solução abrangente para alta disponibilidade, recuperação de desastres e escalabilidade para bancos de dados críticos, permitindo que as empresas protejam seus dados e garantam a continuidade dos negócios em ambientes híbridos.

5. **Backup para o Azure:** O SQL Server oferece a capacidade de fazer backups diretamente para o Azure Storage, facilitando a criação de cópias de segurança na nuvem e garantindo a recuperação de dados em caso de falhas locais.
6. **Integração com Azure Monitor e Azure Security Center:** As soluções híbridas do MSSQL Server permitem a integração com serviços de monitoramento e segurança da Azure, oferecendo visibilidade, controle e proteção abrangentes para os bancos de dados.

Essas soluções híbridas oferecem uma maneira flexível e eficaz de aproveitar o melhor dos ambientes locais e na nuvem, garantindo a escalabilidade, disponibilidade e segurança necessárias para os sistemas de banco de dados corporativos.

Vamos tentar contextualizar um pouco mais com cenários de dois casos de uso com seus benefícios associados das soluções híbridas:

Caso de Uso 1: Empresa de Varejo com Pico de Demanda sazonal

A Empresa é um e-commerce de moda que experimenta picos de vendas durante a temporada de festas. Eles possuem um banco de dados local para gerenciar transações diárias e ações do cliente. Para lidar com o aumento de tráfego durante os períodos de pico, a Empresa A decidiu adotar uma abordagem híbrida. Na nuvem, a Empresa A configura uma instância do Azure SQL Database para armazenar dados históricos e de análise, permitindo que esses dados sejam facilmente acessíveis durante os picos de demanda. Eles também usam o SQL Server Stretch Database para estender dados menos acessados para a nuvem, liberando espaço em seus servidores locais.

Benefícios:

1. **Elasticidade:** Utilizando uma solução híbrida, a empresa pode escalar horizontalmente seus recursos de banco de dados local durante os picos sazonais de demanda, enquanto migra parte de seus dados históricos para a nuvem, mantendo os recursos locais otimizados para o desempenho diário.
2. **Redução de Custos:** Durante os períodos de baixa demanda, os dados menos acessados podem ser movidos para a nuvem, onde os custos de armazenamento podem ser mais econômicos. Isso evita a necessidade de manter recursos locais ociosos.

Caso de Uso 2: Empresa com Requisitos de Recuperação de Desastres



A Empresa é uma consultoria financeira que lida com informações confidenciais dos clientes e requer alta disponibilidade e resiliência. Eles precisam garantir a recuperação rápida em caso de desastres que afetem seus sistemas locais. Para a recuperação de desastres, a Empresa configura Always On Availability Groups com réplicas no Azure para manter cópias sincronizadas de seus dados críticos na nuvem. Isso garante que, em caso de falha do data center local, eles possam ativar rapidamente as réplicas no Azure para manter a operação.

Benefícios:

1. **Recuperação de Desastres:** Ao usar soluções híbridas como Always On Availability Groups com réplicas no Azure, a empresa pode garantir a continuidade dos negócios em caso de falha do data center local, com a capacidade de ativar rapidamente réplicas no Azure para manter a operação.
2. **Economia de Recursos:** Manter réplicas no Azure pode reduzir o investimento em infraestrutura de recuperação de desastres local, permitindo à empresa economizar em custos de hardware e manutenção, além de simplificar os processos de gerenciamento da recuperação de desastres.

Esses casos demonstram como as soluções híbridas podem oferecer benefícios significativos em termos de flexibilidade, escalabilidade, economia de custos e segurança para empresas em diversos cenários operacionais.

Particionamento de Tabela

No SQL Server, é possível dividir o armazenamento de uma tabela ou índice em unidades lógicas chamadas partições. Isso facilita a gestão e a manutenção, tratando a tabela como uma só, mesmo estando dividida em partes. Todas as tabelas no SQL Server são, de alguma forma, particionadas internamente, com uma partição lógica por tabela como padrão.

A ideia por trás disso é conhecida como partição horizontal. Imagine uma tabela de banco de dados que está crescendo demais, tornando lento adicionar novas entradas. Nesse caso, é possível dividir a tabela em grupos com base em uma chave de partição, geralmente uma coluna de data, e armazenar cada grupo em sua própria partição. Essas partições podem ser guardadas em diferentes grupos de arquivos para melhorar o desempenho de leitura e escrita.

Esse tipo de divisão possibilita uma otimização de consultas chamada eliminação de partições. Ela permite que apenas a partição contendo os dados necessários seja consultada. No entanto, vale ressaltar que a partição de tabelas não garante automaticamente um melhor desempenho de consulta; na verdade, em certos casos, o desempenho pode até piorar devido a fatores relacionados a estatísticas.

Apesar disso, há vantagens significativas na divisão de tabelas, especialmente para conjuntos de dados extensos, principalmente em cenários envolvendo janelas temporais e movimentação de dados dentro e fora da tabela. Esse processo é conhecido como troca de partições, que permite mover dados de forma quase instantânea entre a tabela principal e uma tabela separada. Isso facilita o gerenciamento de grandes volumes de dados ou de dados que são atualizados periodicamente.



Quanto à escolha entre tabelas particionadas e visualizações particionadas, embora a partição de tabelas esteja disponível em todas as edições do SQL Server e possa ser atrativa para bancos de dados menores, pode ser mais recomendado optar por visualizações particionadas. Essas visualizações utilizam uma união de consultas em tabelas subjacentes, fornecendo uma alternativa para consulta em vez de acessar diretamente as tabelas particionadas. Utilizar restrições de chave nas tabelas base ainda permite que o otimizador de consultas explore táticas como a "eliminação de partições". Além disso, a movimentação de dados dentro e fora das visualizações particionadas é quase instantânea, uma vez que apenas a própria visualização precisa ser atualizada para adicionar ou remover alguma tabela base específica.

O particionamento de tabelas acontece quando você projeta uma tabela que armazena dados de uma única entidade lógica em estruturas fisicamente separadas. Ou seja, em vez de armazenar todos os dados da entidade em uma única estrutura de dados física, os dados são divididos em múltiplas estruturas físicas, mas o usuário continua a tratá-los como uma unidade única, como de costume.

O particionamento de tabelas tem múltiplos propósitos, alguns dos quais relacionados ao desempenho, seja ao consultar ou ao carregar dados. Existem dois tipos de particionamento: particionamento horizontal e particionamento vertical. Vamos focar nossa atenção no particionamento horizontal que é o mais utilizado.

Particionamento horizontal

| EntityId | A | B | C |
|----------|---|---|---|
| | | | |
| | | | |
| ... | | | |
| | | | |
| | | | |

Como ilustrado na figura acima, o particionamento horizontal divide as linhas de dados, e cada partição possui o mesmo esquema. O particionamento vertical, por outro lado, divide as colunas da entidade em várias tabelas. A figura mostra uma tabela particionada em apenas duas partições, mas é possível particionar tabelas em muitas partições. Você também pode misturar o particionamento horizontal e vertical.

No SQL Server, o particionamento geralmente se refere **ao particionamento horizontal**, somente porque é o nome de um recurso. Em um banco de dados em grande escala, onde uma única tabela pode crescer para centenas de gigabytes ou mais, algumas operações se tornam mais difíceis. Por exemplo, adicionar novas linhas pode levar uma quantidade excessiva de tempo e causar falhas em consultas SELECT na tabela devido à escalada de bloqueio. Preocupações semelhantes existem em relação à remoção de dados e à manutenção de índices.



O **particionamento horizontal** pode lidar com essas preocupações. No entanto, não é uma solução infalível que fará com que todos os problemas de desempenho em tabelas grandes desapareçam. Pelo contrário, quando aplicado incorretamente, o particionamento horizontal pode ter um efeito negativo na carga de trabalho do seu banco de dados.

O suporte para particionamento horizontal no SQL Server era limitado à edição Enterprise do SQL Server até o lançamento do SQL Server 2016 com o Service Pack 1. Desde então, **todas as edições suportam particionamento horizontal de tabelas e índices.**

O recurso de particionamento do SQL Server suporta o particionamento horizontal com **uma função de partição**, que determina em qual partição da tabela uma determinada linha será armazenada. Cada partição pode ser armazenada em seu próprio conjunto de arquivos no mesmo banco de dados.

Ao particionar uma tabela, as linhas da tabela não são todas armazenadas no mesmo local físico. Portanto, ao projetar partições, você deve decidir sobre **uma chave de partição**, que é a coluna que será usada para atribuir uma linha a exatamente uma partição. Do ponto de vista lógico, no entanto, todas as linhas pertencem à mesma tabela.

Uma consulta sem uma cláusula WHERE retorna todas as linhas, independentemente da partição que estão armazenadas. Isso significa que o mecanismo de banco de dados deverá fazer mais trabalho para recuperar linhas de diferentes partições. Seu objetivo ao particionar visando melhoria de desempenho de consulta deve ser escrever consultas que eliminem partições (não façam acesso a determinadas partições). Isso pode ser feito incluindo a chave de partição na cláusula WHERE.

Benefícios adicionais do particionamento horizontal incluem **a capacidade de definir grupos de arquivos específicos como somente leitura**. Ao mapear partições contendo dados mais antigos para grupos de arquivos somente leitura, você pode garantir que esses dados sejam imutáveis sem afetar a capacidade de inserir novas linhas. Além disso, você pode excluir grupos de arquivos somente leitura de backups regulares. Finalmente, durante uma restauração, os grupos de arquivos contendo os dados mais recentes podem ser restaurados primeiro, permitindo que novas transações sejam registradas mais rapidamente do que se todo o banco de dados precisasse ser restaurado.

Além do particionamento de tabelas horizontais, o SQL Server também suporta o particionamento de índice. Um índice particionado é considerado alinhado com a tabela se a tabela e o índice forem particionados no mesmo número de partições usando a mesma coluna e valores de limite.

Quando um índice particionado está alinhado, você pode direcionar operações de manutenção do índice para uma partição específica. Isso pode acelerar significativamente a operação de manutenção, pois você pode reconstruir ou reorganizar uma partição em vez do índice inteiro. Por outro lado, se for necessário reconstruir todo o índice, o SQL Server tentará fazê-lo em paralelo. Reconstruir vários índices simultaneamente cria pressão de memória. Por causa disso, recomenda-se não usar o particionamento em um sistema com menos de 16 GB de RAM.

Para alcançar um índice alinhado, usamos a mesma função de particionamento e esquema da tabela. No entanto, não é estritamente necessário criar um índice particionado alinhado.



É possível se beneficiar da criação um índice particionado sem particionar a tabela. Você ainda pode usar esse índice não alinhado para melhorar a eficiência da consulta se apenas uma ou algumas partições do índice precisarem ser usadas. Nesse caso, você deve usar a chave de partição do índice na cláusula WHERE para obter o benefício de desempenho da eliminação de partições.

A seguir, demonstramos como criar uma tabela particionada horizontalmente usando o recurso do SQL Server. Três objetos do banco de dados estão envolvidos na definição de partições e na partição de uma tabela:

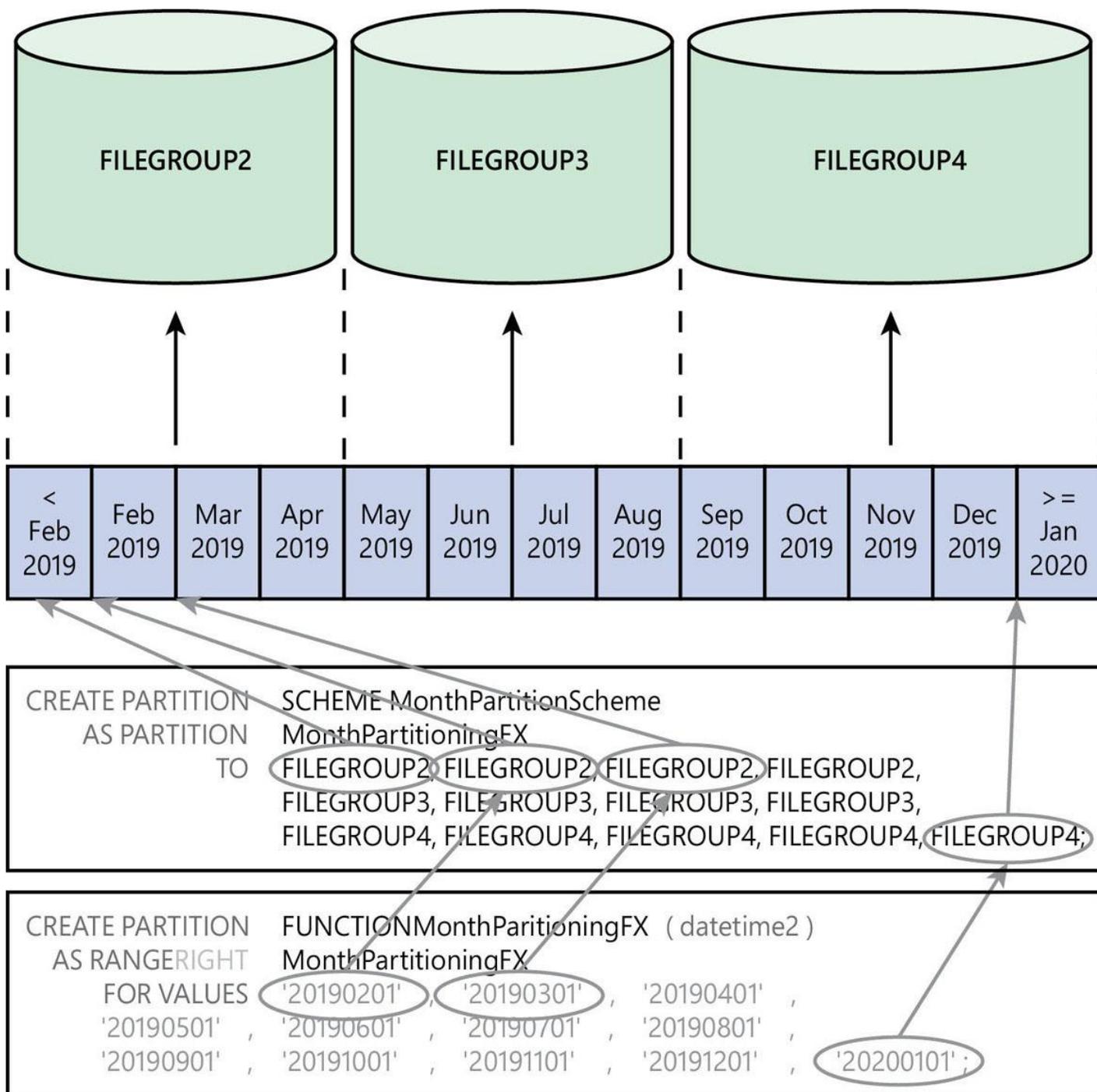
- Uma função de partição, que define o número de partições e os valores de limite
- Um esquema de partição, que define em qual grupo de arquivos cada partição é colocada
- A tabela a ser particionada

Para concisão, o script a seguir não mostra a criação do banco de dados com os grupos de arquivos e arquivos necessários para suportar o esquema de partição.

```
-- Função de partição para 01 de fevereiro de 2019 a 01 de janeiro de 2020
CREATE PARTITION FUNCTION MonthPartitioningFx (datetime2)
-- Valores limite na partição a direita
AS RANGE RIGHT
-- Cada mês é definido pelo seu primeiro dia (o valor de limite)
FOR VALUES ('20190201', '20190301', '20190401', '20190501', '20190601',
'20190701', '20190801', '20190901', '20191001', '20191101', '20191201',
'20200101');
-- Um esquema de partição usando a função de partição
-- Coloca-se cada trimestre em sua própria partição
-- O mês mais recente dos 13 meses vai na última partição
CREATE PARTITION SCHEME MonthPartitioningScheme AS PARTITION
MonthPartitioningFx TO (FILEGROUP2, FILEGROUP2, FILEGROUP2, FILEGROUP2,
FILEGROUP3, FILEGROUP3, FILEGROUP3, FILEGROUP3, FILEGROUP4, FILEGROUP4,
FILEGROUP4, FILEGROUP4, FILEGROUP4);
```

Se você visualizar os dados da tabela como sendo ordenados pela chave de partição em ordem crescente, a partição da esquerda é a partição no topo. Ao definir uma função de partição, você indica se o valor de limite — neste exemplo, o primeiro dia de cada mês — será armazenado na partição à esquerda (o padrão) ou na partição à direita (como especificado no exemplo).





A figura acima mostra a relação entre a função de partição e o esquema de partição. No exemplo, a função de partição criou 13 partições usando 12 valores de limite. O esquema de partição então direcionou essas 13 partições para três grupos de arquivos, especificando cada grupo de arquivos quatro vezes e o último grupo de arquivos cinco vezes (porque ele conterá a última partição).

Diante da imagem, visualiza-se a relação entre **a função de partição, o esquema de partição e os grupos de arquivos** nos quais as partições serão armazenadas. Um exemplo de código usado para criar a função de partição é mostrado na parte inferior. O código exibido é:



```
CREATE PARTITION FUNCTION MonthPartitioningFx (datetime2) AS RANGERIGHT  
FOR VALUES ('20190201', '20190301', '20190401', '20190501', '20190601',  
'20190701', '20190801', '20190901', '20191001', '20191101', '20191201',  
'20200101');
```

Diretamente acima deste código, há outro exemplo de código para criar o esquema de partição. O código exibido é o seguinte:

```
CREATE PARTITION SCHEME MonthPartitioningScheme AS PARTITION  
MonthPartitioningFx TO FILEGROUP2, FILEGROUP2, FILEGROUP2, FILEGROUP2,  
FILEGROUP3, FILEGROUP3, FILEGROUP3, FILEGROUP3, FILEGROUP3, FILEGROUP4,  
FILEGROUP4, FILEGROUP4, FILEGROUP4, FILEGROUP4;
```

No topo do diagrama, há três imagens em formato de banco de dados representando os três diferentes grupos de arquivos, FILEGROUP2, FILEGROUP3 e FILEGROUP4. Diretamente abaixo das imagens dos grupos de arquivos, há várias caixas uma ao lado da outra representando os intervalos de datas. Para FILEGROUP2, há quatro caixas < fev 2019, fev 2019, mar 2019 e abr 2019.

Para FILEGROUP 3, também há quatro caixas, maio 2019, jun 2019, jul 2019 e ago 2019. FILEGROUP 4 possui cinco caixas, set 2019, out 2019, nov 2019, dez 2019 e >= jan 2020. A primeira ocorrência de FILEGROUP2, no esquema de partição, vai para o início da caixa < fev 2019. Nenhum valor é mostrado para esses dados na função de partição.

O diagrama também mostra como o intervalo de data '20190201' da função de partição está relacionado com a segunda ocorrência de FILEGROUP2 no esquema de partição. Este esquema de partição está mostrado relacionado com o início da caixa fev 2019, abaixo de FILEGROUP2. Da mesma forma, o intervalo de data '20190301' na função de partição é exibido na terceira ocorrência de FILEGROUP2 no esquema de partição. O FILEGROUP2 do esquema de partição aponta para o início da caixa mar 2019 acima. A data '20200101' na função de partição está relacionada com a última entrada FILEGROUP4 no esquema de partição. Este esquema de partição vai para o início da caixa mais à direita, para dados maiores ou iguais a 1 de jan de 2020.

Diretrizes de design para o particionamento horizontal

Ao projetar partições horizontais, tenha em mente as seguintes diretrizes, com o entendimento de que os resultados podem variar:

- O número de operações paralelas que podem ser executadas por consulta depende do número de núcleos do processador no sistema. Usar mais partições que núcleos de processador limita o número de partições que serão processadas em paralelo. Portanto, mesmo que o SQL Server agora suporte até 15.000 partições, em um sistema com 12 núcleos de processador, no máximo 12 partições serão processadas em paralelo. Você pode escolher usar menos partições do que o número de núcleos de processador disponíveis para reservar capacidade para outras consultas. Há também a opção de desativar o processamento paralelo de partições ou alterar o número de processadores que uma única consulta pode utilizar. Esteja ciente de que alterar o número de processadores por consulta é uma configuração do servidor.



- Escolha a chave de partição para acomodar valores de coluna em crescimento. Isso poderia ser um valor de data ou uma coluna de identidade incremental. Normalmente, você vai querer que novas linhas sejam adicionadas à partição mais à direita.
- A chave de partição selecionada deve ser imutável, ou seja, não deve haver motivo de negócios para que o valor dessa chave seja alterado. Se o valor de uma chave de partição mudar, o SQL Server executará a declaração UPDATE como uma declaração DELETE e INSERT; não há provisão para "mover" uma linha para outro grupo de arquivos. Esse procedimento é similar à situação em que o valor de um índice clusterizado é alterado.
- Para a chave de partição, um tipo de dados estreitos é preferível em relação a um tipo de dados amplo.
- Para obter a maioria dos benefícios do particionamento, especificamente aqueles relacionados ao desempenho, você precisará colocar cada partição em seu próprio grupo de arquivos. Isso não é um requisito, e algumas ou todas as partições podem compartilhar um único grupo de arquivos. Por exemplo, a próxima seção discute uma estratégia de particionamento de janela deslizante, em que o particionamento é benéfico mesmo se todas estiverem no mesmo grupo de arquivos.
- Considere o armazenamento que sustenta os grupos de arquivos. Por exemplo, seu sistema de armazenamento pode não fornecer maior desempenho se todos os grupos de arquivos estiverem em unidades físicas diferentes. Esteja ciente de que, mesmo que as letras das unidades sejam diferentes, elas ainda podem estar todas na mesma unidade física.
- As tabelas que são bons candidatos para particionamento são tabelas com muitas — em milhões ou bilhões — linhas para as quais os dados são principalmente adicionados em vez de atualizados, e nas quais as consultas são frequentemente executadas para retornar dados de uma ou algumas partições.

Algumas vezes é necessário implementar **uma estratégia de particionamento de janela deslizante**. O particionamento horizontal é frequentemente aplicado a data warehouses relacionais. Uma operação comum de um data warehouse é carregar uma quantidade significativa de dados em uma tabela de fatos enquanto simultaneamente elimina dados antigos. A estratégia de particionamento de janela deslizante é particularmente adequada para **tabelas em que os dados são adicionados e removidos regularmente**. Por exemplo, os dados em uma tabela de fatos podem ser eliminados após 13 meses. Talvez cada vez que dados são carregados no data warehouse, linhas mais antigas que 13 meses são removidas enquanto novas linhas são adicionadas. Isso é uma janela deslizante, pois a tabela de fatos sempre contém os 13 meses mais recentes de dados.

Para configurar uma janela deslizante, é preciso de uma função de partição e esquema, bem como da tabela de fatos. Você também deve configurar um procedimento armazenado que modifica a função de partição para acomodar os novos valores de limite. Por fim, você precisará de uma tabela de staging com as mesmas colunas e índice clusterizado da tabela particionada.

No SQL Server, o particionamento de tabelas é realizado principalmente para melhorar o desempenho de consultas, facilitar o gerenciamento de grandes volumes de dados e permitir a movimentação de dados de forma mais eficiente. As partições podem ser baseadas em uma coluna específica (como data) e são divididas em segmentos menores, o que acelera operações como exclusões, consultas e manutenção de índices.



Particionamento de Tabelas em Ambientes de Nuvem:

- O particionamento em ambientes de nuvem, como o Azure SQL Database, segue os mesmos princípios do SQL Server local, mas com algumas considerações específicas.
- Em certos serviços de nuvem, o particionamento pode ser gerenciado automaticamente pelo provedor, otimizando o desempenho e a escalabilidade da distribuição dos dados.
- Além disso, em ambientes de nuvem, o particionamento pode ser integrado com outras funcionalidades nativas de nuvem, como o armazenamento de dados em Data Lakes ou aplicação de políticas de segurança específicas da nuvem.

Diferenças no Particionamento em Nuvem:

- Em ambientes de nuvem, o particionamento pode estar mais orientado para a escalabilidade automática e a otimização do desempenho dos serviços de banco de dados específicos da nuvem, adaptando-se às necessidades dinâmicas do ambiente em nuvem.
- As considerações de segurança e conformidade específicas da nuvem também podem influenciar a maneira como o particionamento é configurado e gerenciado em comparação com ambientes locais.

Em resumo, embora os conceitos básicos de particionamento de tabelas sejam semelhantes entre o SQL Server local e o particionamento em ambientes de nuvem, as abordagens específicas e a integração com recursos nativos de nuvem podem variar, de acordo com a dinâmica e os recursos oferecidos pelos serviços de nuvem específicos.

Estratégias de Migração Offline ou Online

A migração de dados offline e online no SQL Server e no Azure apresentam diferenças significativas em relação ao processo e às possibilidades de execução.

A migração offline envolve a interrupção do serviço durante a transferência de dados. Nesse método, o banco de dados é geralmente colocado em modo somente leitura ou totalmente offline, o que significa que não podem ser feitas alterações enquanto a migração está em andamento. Apesar de ser mais simples de ser realizada, a migração offline pode implicar em períodos de inatividade para os usuários, afetando a disponibilidade do sistema.

Por outro lado, a migração online permite que o sistema permaneça operacional durante o processo de transferência de dados. Com esse método, as alterações feitas no banco de dados não são interrompidas, garantindo a continuidade do serviço. A migração online é mais complexa de ser executada, pois exige tecnologias que possibilitem a sincronização dos dados antigos com os novos.

Ao migrar para o Azure, é importante considerar as diferenças entre as migrações offline e online, pois o ambiente de nuvem oferece recursos específicos para cada abordagem. A migração offline pode ser mais



adequada para ambientes controlados e com pouca demanda para disponibilidade contínua, enquanto a migração online é ideal para sistemas que necessitam estar sempre disponíveis.

A Microsoft oferece uma variedade de ferramentas e soluções para auxiliar na migração para versões mais recentes do SQL Server ou para a oferta Azure IaaS ou PaaS. Vamos explorar algumas delas e como podem ser aplicadas em diferentes cenários:

1. Ferramenta de Avaliação e Planejamento da Microsoft (MAP):

- **Descrição:** Um ponto de partida crucial para muitos projetos de migração, o MAP é uma ferramenta multiproduto sem agentes, utilizada para avaliar e planejar a migração.
- **Funcionalidades:**
 - Inventaria toda a infraestrutura de TI, incluindo SQL Server, através da varredura de endereços IP, Active Directory (AD) e ativos de rede.
 - Coleta informações em um banco de dados SQL Server.
 - Possui um conjunto leve de relatórios embutidos.
- **Benefícios:**
 - Oferece uma visão abrangente da sua infraestrutura de banco de dados.
 - Facilita o planejamento e a estimativa de custos da migração.
 - Gera relatórios pré-construídos para auxiliar na tomada de decisões.

2. Calculadora de Custo Total de Propriedade (TCO):

- **Descrição:** Esta ferramenta online da Microsoft ajuda a calcular o TCO de migrações para a Azure, incluindo economias de custo potenciais.
- **Funcionalidades:**
 - Permite definir diferentes cenários de migração e configurações de infraestrutura.
 - Calcula os custos de infraestrutura, licenciamento, software e outros itens relevantes.
 - Apresenta um detalhamento dos custos previstos e das economias potenciais.
- **Benefícios:**
 - Fornece uma estimativa precisa dos custos da migração para a Azure.
 - Permite comparar diferentes cenários e tomar decisões informadas sobre o investimento em migração.
 - Auxilia na justificativa do ROI da migração para stakeholders.

3. Assistente de Experimentação de Banco de Dados (DEA):

- **Descrição:** Essa ferramenta de experimentação auxilia na avaliação de possíveis problemas de atualização do SQL Server.
- **Funcionalidades:**
 - Analisa o banco de dados de origem para identificar problemas de compatibilidade com a nova versão do SQL Server.
 - Fornece relatórios detalhados sobre os problemas encontrados e sugestões para corrigi-los.
 - Permite testar diferentes cenários de atualização em um ambiente seguro.
- **Benefícios:**



- Reduz o risco de problemas durante a atualização do SQL Server.
- Facilita a identificação e correção de problemas de compatibilidade.
- Permite testar a atualização em um ambiente controlado antes da implementação em produção.

4. Assistente de Migração de Dados Azure (DMA):

- **Descrição:** Essa ferramenta auxilia na migração de dados para plataformas de dados modernas, como o Azure SQL Database.
- **Funcionalidades:**
 - Detecta problemas de compatibilidade que podem afetar a funcionalidade do banco de dados na nova plataforma.
 - Sugere soluções para corrigir os problemas de compatibilidade identificados.
 - Gera scripts para automatizar o processo de migração de dados.
- **Benefícios:**
 - Simplifica o processo de migração de dados para o Azure SQL Database.
 - Reduz o risco de erros e problemas de compatibilidade.
 - Automatiza tarefas repetitivas, economizando tempo e esforço.

5. Serviço de Migração de Banco de Dados Azure (DMS):

- **Descrição:** Um serviço totalmente gerenciado para realizar migrações online e offline para plataformas Azure SQL.
- **Funcionalidades:**
 - Migra dados entre diversos bancos de dados, incluindo SQL Server, Oracle, MySQL e PostgreSQL.
 - Suporta migrações online com tempo de inatividade mínimo ou migrações offline para maior controle.
 - Oferece opções de migração incremental e de replicação contínua.
- **Benefícios:**
 - Migrações seguras e confiáveis com mínimo impacto nos aplicativos.
 - Flexibilidade para escolher o tipo de migração que melhor atende às suas necessidades.
 - Gerenciamento completo do processo de migração pela Microsoft.

6. Assistente de Migração de SQL Server (SSMA)

O Assistente de Migração de SQL Server (SSMA) é uma ferramenta robusta da Microsoft projetada para automatizar a migração de bancos de dados para o SQL Server a partir de diversas plataformas, como Microsoft Access, DB2, MySQL, Oracle e SAP ASE. Ele simplifica o processo de migração, reduzindo o tempo e o esforço necessários, e garante a precisão e confiabilidade da conversão de dados e estruturas.

Funcionalidades Essenciais:

- **Conversão de Banco de Dados:** O SSMA converte automaticamente o schema e os dados do banco de dados de origem para o formato compatível com o SQL Server.



- **Gerenciamento de Objetos:** A ferramenta mapeia objetos de banco de dados de origem para objetos equivalentes no SQL Server, garantindo a preservação da lógica e funcionalidade do banco de dados.
- **Validação de Regras de Negócio:** O SSMA verifica se as regras de negócio e restrições do banco de dados de origem serão corretamente aplicadas no SQL Server, evitando problemas de compatibilidade.
- **Geração de Scripts:** A ferramenta gera scripts SQL para executar a migração de dados e schema, facilitando a implementação e o controle do processo.
- **Suporte a Migrações Complexas:** O SSMA oferece recursos avançados para lidar com migrações complexas, como conversão de tipos de dados, triggers, stored procedures e funções.

Benefícios da Migração com SSMA:

- **Agilidade e Eficiência:** Automatiza a migração, reduzindo drasticamente o tempo e o esforço manual necessários.
- **Precisão e Confiabilidade:** Garante a conversão precisa de dados e estruturas, minimizando o risco de erros e problemas de compatibilidade.
- **Redução de Custos:** Elimina a necessidade de reescrever manualmente o código e garante uma migração suave e eficiente.
- **Minimização de Tempo de Inatividade:** Permite migrações online com tempo de inatividade mínimo, minimizando o impacto nos negócios.
- **Flexibilidade e Escalabilidade:** Suporta migrações de bancos de dados de diversos tamanhos e complexidades, adaptando-se às suas necessidades.

Cenários Ideais para Migração com SSMA:

- **Migração de Bancos de Dados Legados:** Modernize seus bancos de dados antigos para o SQL Server, aproveitando seus recursos avançados e segurança aprimorada.
- **Migração para Ambientes Híbridos ou Multicloud:** Mova seus bancos de dados para o SQL Server em ambientes híbridos ou multicloud, aumentando a flexibilidade e escalabilidade.
- **Consolidação de Bancos de Dados:** Combine vários bancos de dados em um único ambiente SQL Server, simplificando o gerenciamento e a manutenção.
- **Atualização para Versões Recentes do SQL Server:** Atualize seus bancos de dados para as últimas versões do SQL Server, beneficiando-se de novos recursos e otimizações.

7. Kit de Migração de Acesso a Dados (DAMT):

- **Descrição:** Uma extensão do Azure Data Studio (ADS) que facilita a migração de código-fonte de aplicativos de uma plataforma de banco de dados para outra.
- **Funcionalidades:**
 - Converte código-fonte de T-SQL, PL/SQL, Oracle PL/SQL e PostgreSQL PL/pgSQL para o formato compatível com a plataforma de destino.
 - Identifica e corrige problemas de compatibilidade de código.
 - Gera scripts de migração para automatizar o processo de migração de código.
- **Benefícios:**



- Simplifica a migração de aplicativos de banco de dados para diferentes plataformas.
- Reduz a necessidade de modificações manuais no código.
- Automatiza tarefas repetitivas, economizando tempo e esforço.

8. Link do Azure SQL Managed Instance para SQL Server:

- **Descrição:** Um novo recurso do SQL Server 2022 e SQL Managed Instance que oferece alta disponibilidade híbrida (HA), recuperação de desastres (DR) e réplicas secundárias legíveis na nuvem.
- **Funcionalidades:**
 - Permite replicar bancos de dados SQL Server locais para uma instância SQL gerenciada no Azure para HA e DR.
 - Oferece réplicas secundárias legíveis na nuvem para consultas e relatórios.
 - Simplifica a configuração e o gerenciamento de HA e DR.
- **Benefícios:**
 - Protege seus dados contra falhas de hardware e software.
 - Permite a recuperação rápida de desastres com impacto mínimo nos negócios.
 - Possibilita análises e relatórios em tempo real em réplicas secundárias na nuvem.

Observações:

- A escolha da ferramenta ideal depende das suas necessidades específicas, tipo de banco de dados de origem e destino, volume de dados, orçamento e tempo disponível para a migração.
- É recomendável avaliar cuidadosamente cada ferramenta e consultar especialistas em migração de banco de dados para obter orientação e suporte.

As ferramentas de migração e modernização de banco de dados SQL da Microsoft oferecem soluções abrangentes para auxiliar empresas na jornada de migração para plataformas modernas. Ao escolher a ferramenta correta e utilizar as melhores práticas, você pode garantir uma migração suave, eficiente e segura, maximizando os benefícios da modernização do seu banco de dados.

Sincronização de Dados SQL para Azure

A sincronização de dados é o processo de garantir que as informações em diferentes fontes de dados estejam atualizadas e consistentes. No contexto da migração e integração de dados no Azure, a sincronização de dados desempenha um papel crucial ao permitir que bancos de dados em locais distintos estejam alinhados e sempre atualizados.

A importância da sincronização de dados no Azure pode ser destacada pelos seguintes pontos:

1. **Consistência dos dados:** Ao sincronizar dados entre diferentes fontes, é possível garantir que as informações sejam consistentes e precisas em todo o ambiente de dados, independentemente de sua localização.



2. **Evitar inconsistências:** A sincronização de dados ajuda a evitar inconsistências e conflitos que podem surgir quando múltiplas instâncias de um banco de dados são atualizadas separadamente, garantindo a integridade dos dados.
3. **Requisitos de disponibilidade:** Em ambientes distribuídos, onde a disponibilidade e o acesso rápido aos dados são essenciais, a sincronização de dados no Azure permite manter os dados atualizados em tempo real em diferentes localizações geográficas.
4. **Suporte à migração de dados:** Durante processos de migração de dados para o Azure, a sincronização de dados pode ser utilizada para garantir uma transição suave, mantendo os dados consistentes entre os sistemas legados e os novos ambientes na nuvem.
5. **Integração de sistemas:** Ao integrar diferentes sistemas e fontes de dados no Azure, a sincronização de dados facilita o compartilhamento e a atualização de informações entre esses sistemas, permitindo uma visão unificada dos dados.

O **Azure SQL Data Sync** é um serviço crucial para empresas que gerenciam dados em múltiplos bancos de dados, tanto em ambientes locais quanto na nuvem. Essa ferramenta poderosa permite a sincronização bidirecional de dados entre bancos de dados SQL Server e Azure SQL Database, garantindo que suas informações estejam sempre consistentes e atualizadas em todas as localidades. Neste sentido, podemos listar os seguintes cenários de uso:

- **Sincronização Híbrida de Dados:** Mantenha seus dados em sincronia entre bancos de dados SQL Server locais e Azure SQL Database, facilitando a migração para a nuvem e a implementação de aplicações híbridas.
- **Aplicações Distribuídas:** Sincronize dados entre diferentes bancos de dados para workloads específicos, como separar bancos de dados de produção de bancos de dados de relatórios e análises, otimizando o desempenho e a escalabilidade.
- **Aplicações Globalmente Distribuídas:** Mantenha a consistência de dados em bancos de dados SQL espalhados por diferentes regiões, minimizando a latência de rede e atendendo às necessidades de negócios em diversas localidades.

Diferenças entre Sincronização de Dados SQL do Azure e Replicação Transacional:

| Característica | Sincronização de Dados SQL do Azure | Replicação Transacional |
|--|-------------------------------------|-------------------------|
| Suporte Ativo-Ativo | Sim | Sim |
| Sincronização Bidirecional (Local e Azure) | Sim | Sim |
| Latência | Menor | Maior |
| Consistência Transacional | Não | Sim |
| Suporte ao Azure SQL Managed Instance | Não | Sim |



| | | |
|-----------------------|-------|-------|
| Impacto no Desempenho | Maior | Menor |
| Custo de Manutenção | Menor | Maior |

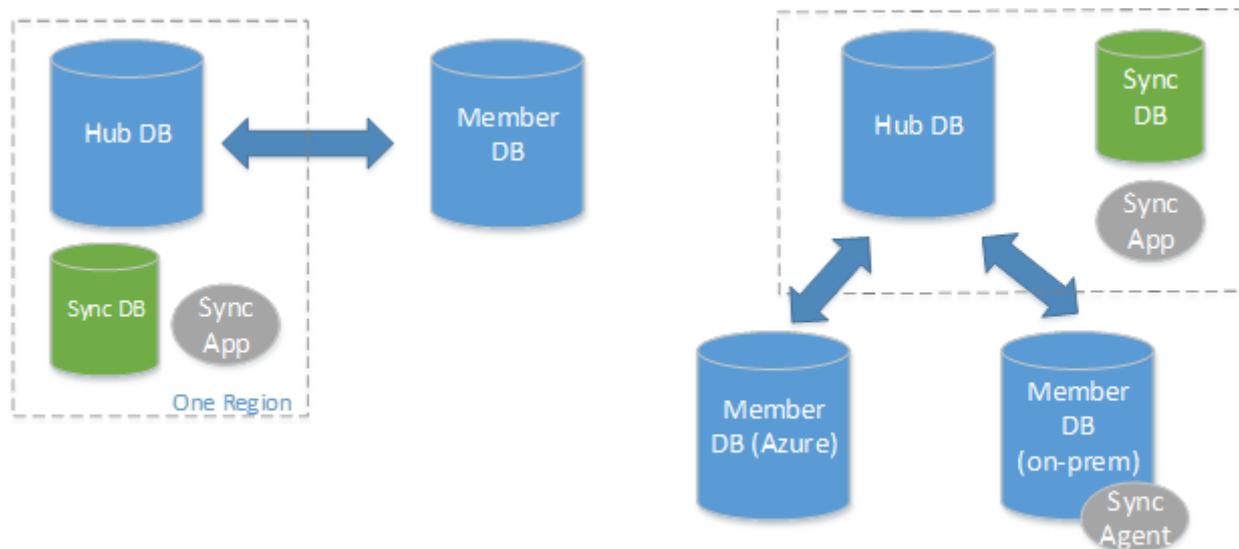


Figura 1 - Sincronização entre 2 bancos de dados (esquerda) e entre 3 bases (direita)

Funcionamento Detalhado:

- Rastreamento de Alterações de Dados:** A Sincronização de Dados SQL do Azure utiliza triggers de inserção, atualização e exclusão para monitorar alterações nos dados. Essas alterações são registradas em uma tabela auxiliar dentro do banco de dados do usuário. É importante notar que a instrução BULK INSERT não aciona triggers por padrão. Para que a Sincronização de Dados SQL do Azure possa rastrear essas inserções, é necessário especificar a opção FIRE_TRIGGERS.
- Sincronização de Dados:** A Sincronização de Dados SQL do Azure adota uma topologia hub-and-spoke, onde o hub sincroniza individualmente com cada membro. As alterações do hub são baixadas para o membro e, em seguida, as alterações do membro são carregadas para o hub.
- Resolução de Conflitos:** A Sincronização de Dados SQL do Azure oferece duas opções para resolução de conflitos: **Hub** ou **Member**.
 - Hub vence:** As alterações no hub sempre sobrescrevem as alterações no membro, garantindo que a versão do hub seja a fonte de verdade.
 - Member vence:** As alterações no membro sobrescrevem as alterações no hub. Se houver mais de um membro, o valor final dependerá de qual membro sincronizar primeiro.

Benefícios Adicionais:



- **Flexibilidade:** Suporta diversos tipos de bancos de dados, incluindo Azure SQL Database, SQL Server em diferentes versões e até mesmo Oracle, MySQL e PostgreSQL.
- **Escalabilidade:** Adapta-se às suas necessidades de sincronização, desde pequenos conjuntos de dados até grandes bancos de dados com alta frequência de alterações.
- **Segurança:** Protege seus dados durante a sincronização com criptografia e autenticação robustas.
- **Monitoramento:** Permite o monitoramento detalhado do status da sincronização, logs e recursos do sistema, garantindo a confiabilidade do processo.

O Azure SQL Data Sync oferece flexibilidade na sincronização de dados entre bancos de dados, supondo diferentes tipos de topologias e direcionamentos de alterações. Compreender cada tipo e as melhores práticas para sua seleção é crucial para garantir a eficiência e confiabilidade da sincronização em diversos cenários.

Tipos de Sincronização:

1. Sincronização Bidirecional:

- **Descrição:** Permite que alterações em **qualquer** banco de dados do grupo de sincronização sejam propagadas para todos os outros bancos de dados.
- **Cenários Ideais:**
 - Quando a consistência de dados em tempo real entre todos os bancos de dados é crucial.
 - Colaboração entre equipes que modificam dados em diferentes bancos de dados.
- **Melhores Práticas:**
 - Utilize em cenários com alta frequência de alterações e baixa latência tolerável.
 - Implemente mecanismos de controle de conflito para resolver conflitos de alterações simultâneas.

2. Sincronização Unidirecional:

- **Descrição:** Permite que alterações em um banco de dados de **origem** sejam propagadas para um ou mais bancos de dados de **destino**.
- **Cenários Ideais:**
 - Replicação de dados de um banco de dados central para filiais ou backups.
 - Publicação de dados somente leitura para relatórios ou análises.
- **Melhores Práticas:**



- Utilize em cenários com fluxo unidirecional de dados e baixa necessidade de sincronização em tempo real.
- Otimize a topologia para minimizar o tráfego de rede e a carga nos bancos de dados de destino.

3. Topologias de Sincronização:

○ Hub-and-Spoke:

- **Descrição:** Um banco de dados central (hub) sincroniza com diversos bancos de dados periféricos (spokes).
- **Cenários Ideais:**
 - Sincronização de dados de um escritório central para filiais.
 - Agregação de dados de diversas fontes para um data warehouse central.
- **Melhores Práticas:**
 - Escolha um servidor potente para o hub que suporte a carga de processamento da sincronização.
 - Implemente balanceamento de carga para distribuir o tráfego entre os spokes.

○ Mesh:

- **Descrição:** Cada banco de dados no grupo sincroniza com **todos** os outros bancos de dados do grupo.
- **Cenários Ideais:**
 - Colaboração entre equipes que modificam dados em diferentes bancos de dados sem um único ponto central de controle.
 - Distribuição de dados entre diversos servidores para alta disponibilidade e escalabilidade.
- **Melhores Práticas:**
 - Utilize em cenários com topologia complexa e necessidade de redundância robusta.
 - Monitore de perto o desempenho e o consumo de recursos para evitar gargalos.

Considerações Adicionais:



- **Latência:** A latência da sincronização depende do tipo de sincronização, topologia, volume de dados e recursos da rede.
- **Conflito de Alterações:** Mecanismos de resolução de conflitos devem ser configurados para lidar com alterações simultâneas em dados sincronizados.
- **Segurança:** Utilize criptografia e autenticação para proteger dados durante a sincronização.
- **Monitoramento:** Monitore o status da sincronização, logs e recursos para garantir o bom funcionamento do processo.

Ao escolher o tipo de sincronização e topologia, avalie cuidadosamente seus requisitos de negócio, padrões de acesso a dados, tolerância à latência e infraestrutura disponível. Utilize as melhores práticas mencionadas para garantir uma sincronização eficiente, confiável e segura.



QUESTÕES COMENTADAS - CESPE (CEBRASPE)

1. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A unidade mais comum de medida de área é o hectare.

Gabarito: Errado

Comentário: O item apresenta uma informação incorreta. O hectare não é a unidade mais comum de medida de área; é uma unidade de medida de área, mas não é a mais comum. A unidade mais comum de medida de área depende do contexto geográfico e das práticas locais.

As unidades mais comuns de medida de área podem incluir metros quadrados (m²) e quilômetros quadrados (km²) em sistemas métricos, acres em alguns contextos anglo-americanos, entre outras unidades regionais.

Portanto, a afirmação de que o hectare é a unidade mais comum de medida de área é incorreta. A escolha da unidade de medida de área depende do sistema de unidades adotado e das práticas locais.

2. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A função ST_Area permite calcular a área de polígonos.

Gabarito: Certo

Comentário: A afirmação está correta. No SQL Server, a função ST_Area faz parte das funcionalidades relacionadas à extensão espacial (Spatial Extensions) e é usada para calcular a área de polígonos em um sistema de referência espacial (SRID) específico. A sintaxe geral da função ST_Area seria algo assim:

```
SELECT geometry_column.ST_Area() AS Area FROM your_table;
```

Essa consulta retornaria a área do polígono representado pela coluna geometry_column na tabela your_table. É importante observar que o resultado da área dependerá da unidade de medida do sistema de referência espacial utilizado. Portanto, se você precisa calcular a área de polígonos em um contexto espacial no SQL Server, a função ST_Area é apropriada para essa finalidade.

3. CEBRASPE (CESPE) - Especialista Técnico (BNB)/Analista de Sistema/2018



Acerca de bancos de dados, julgue o item que se segue.

O código a seguir, criado no SQL Server 2017, apresenta uma visão materializada, especificamente devido ao argumento SCHEMABINDING.

```
CREATE VIEW
```

```
VwTeste WITH
```

```
SCHEMABINDING
```

```
AS
```

```
SELECT campo1 FROM tabela WHERE campo1 > 17;
```

Comentário: Lembre-se que o parâmetro SCHEMABINDING serve para criar uma dependência entre as views e o esquemas do banco de dados que foi usado para a criação da visão. Ele impede que modificações feitas no esquema inviabilizem o uso da visão.

Gabarito: E

4. CEBRASPE (CESPE) - Professor de Educação Básica (SEDF)/Informática/2017

Julgue o item a seguir, a respeito de banco de dados, organização de arquivos, métodos de acesso e banco de dados textuais.

Ao executar consultas aninhadas, os bancos de dados SQL Server e DB2 utilizam avaliação correlacionada para eliminar as correlações, sem considerarem como opção o nivelamento das consultas aninhadas ou a utilização de técnicas de reescrita.

Comentário: O texto desta questão é meio sem fundamento. Veja que a consulta correlacionada funciona como um encadeamento de loop e a consulta interna usa os elementos da consulta externa, isso tem que acontecer sempre, não existe mágica que SQL Server possa fazer para eliminar a correlação no processamento. Logo, temos uma alternativa incorreta.

Gabarito: E

5. CEBRASPE (CESPE) - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

A ferramenta SQL Server Configuration Manager permite realizar configurações de modo que uma instância do SQL Server se inicie automaticamente quando o servidor for ligado.

Comentário: O SQL Server Configuration Manager é uma ferramenta para gerenciar os serviços associados ao SQL Server, server para configurar os protocolos de rede usados pelo SQL Server e para gerenciar a configuração de conectividade de rede de computadores cliente do SQL Server. A ferramenta é instalada em conjunto com o SQL Server.

Durante a instalação, o SQL Server normalmente é configurado para iniciar automaticamente. Se isto não foi feito, você poderá alterar essa definição a qualquer momento usando o SQL Server Configuration Manager.



Gabarito: C

6. CEBRASPE (CESPE) - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

Caso a senha de uma conta do SQL Server 2008 R2 seja alterada, a nova senha entrará em vigor imediatamente, sem a necessidade de reinicialização do SQL Server.

Comentário: O Mecanismo de Banco de Dados do SQL Server e o SQL Server Agent são executados em um computador como um serviço usando credenciais fornecidas inicialmente durante a instalação. Se a instância do SQL Server estiver sendo executada na conta de domínio e a senha para aquela conta for alterada, a senha usada pelo SQL Server deverá ser atualizada para a senha nova. Se a senha não for atualizada, o SQL Server poderá perder acesso a alguns recursos de domínio e se o SQL Server parar, o serviço não será reinicializado até que a senha seja atualizada.

Em uma instância autônoma do SQL Server, a senha entra em vigor imediatamente, sem reinicializar o SQL Server. Uma informação complementar que não foi abordada na questão: em uma instância clusterizada, o SQL Server poderia usar o recurso do SQL Server offline, e exigir uma reinicialização.

Gabarito: C.

7. Ano: 2015 Banca: CESPE Órgão: STJ Prova: Analista Judiciário - Infraestrutura

A respeito da configuração e administração de banco de dados, julgue os próximos itens.

[1] Diferentemente das versões anteriores, o SQL Server 2014 não pode ser instalado em computadores com sistema de arquivos FAT32, mas apenas em computadores com sistema de arquivos NTFS.

Comentário: Os requisitos de espaço em disco variam de acordo com os componentes do SQL Server instalados. É recomendado instalar o SQL Server em computadores com os formatos de arquivo NTFS ou ReFS. O sistema de arquivos FAT32 tem suporte, mas não é recomendado, pois é menos seguro do que demais sistemas de arquivos (NTFS ou ReFS).

Curiosidade: Unidades somente leitura, mapeadas ou compactadas são bloqueadas durante a instalação do SQL Server.

Gabarito: E.

8. CEBRASPE (CESPE) - Inspetor de Controle Externo (TCE-RN)/Tecnologia da Informação/2015

No que se refere a tecnologia e arquitetura de banco de dados, julgue o próximo item.

No MSSQL Server 2014, o recurso AlwaysOn é uma solução de alta disponibilidade e de recuperação de desastres que fornece uma alternativa, em nível corporativo, para o espelhamento de bancos de dados, a partir do gerenciamento de réplicas de bancos



de dados.

Comentário: AlwaysOn é um termo genérico para os recursos de disponibilidade do SQL Server e aborda os grupos de disponibilidade e FCIs.

Introduzidos no SQL Server 2012, os Grupos de Disponibilidade AlwaysOn fornecem proteção em nível de banco de dados ao enviar cada transação de um banco de dados para outra instância, conhecida como uma réplica, que contém uma cópia do banco de dados em um estado especial.

Um grupo de disponibilidade pode ser implantado nas versões Standard ou Enterprise Editions. As instâncias que participam de um grupo de disponibilidade podem ser autônomas ou Instâncias de Cluster de Failover AlwaysOn (FCIs - Failover Cluster Instance. Como as transações são enviadas a uma réplica conforme acontecem, os grupos de disponibilidade são recomendados onde houver requisitos de recuperação imediata.

A movimentação de dados entre as réplicas pode ser síncrona ou assíncrona, com a Enterprise Edition, permitindo que até três réplicas (inclusive a primária) sejam síncronas. Um grupo de disponibilidade tem uma cópia de leitura/gravação completa do banco de dados que está na réplica primária, enquanto todas as réplicas secundárias não podem receber transações diretamente de usuários finais ou de aplicativos.

Gabarito: C.

9. Ano: 2015 Banca: CESPE Órgão: MEC Prova: TÉCNICO DE NÍVEL SUPERIOR - ANALISTA DE SISTEMAS

Acerca dos sistemas gerenciadores de banco de dados (SGBD) PostgreSQL, Microsoft SQL Server e Oracle, julgue os itens a seguir.

Uma das principais novidades do Microsoft SQL Server 2014 é o recurso OLTP na memória (In- memory OLTP), o qual permite melhorar significativamente o desempenho de sistemas com processamento de transações on-line e data warehousing. A única maneira de remover um grupo de arquivos com otimização de memória é descartar o banco de dados.

Comentário: O OLTP in-memory é a principal tecnologia disponível no SQL Server e Banco de Dados SQL para otimizar o desempenho do processamento de transações, ingestão de dados, carregamento de dados e cenários de dados transitórios.

Em essência, o OLTP in-memory melhora o desempenho de processamento, tornando a execução de transações e o acesso aos dados mais eficaz, removendo a contenção de bloqueio e de trava entre transações em execução simultânea: não é rápido porque ele está na memória. é rápido porque ele é otimizado em torno de dados na memória. Os algoritmos de processamento, o acesso e o armazenamento de dados foram reprojatados desde o início para tirar proveito dos aprimoramentos mais recentes da computação na memória e de alta simultaneidade.

Antes de começar a usar o OLTP in-memory, você precisa criar um grupo de arquivos (filegroup) do tipo MEMORY_OPTIMIZED_DATA. A única maneira de remover um grupo de arquivos com otimização de memória é descartar o banco de dados.

Gabarito: C



10. CEBRASPE (CESPE) - Analista Judiciário (TJDFT)/Apoio Especializado/Suporte em Tecnologia da Informação/2015

A respeito da configuração e da administração de sistemas gerenciadores de bancos de dados (SGBD) e de produtos a eles relacionados, julgue o item a seguir.

Se, na modificação de determinada instância existente de SQL Server, for realizada a instalação de componentes de replicação, será necessário reiniciar o agente de SQL Server.

Comentário: Segundo a Microsoft, é possível instalar componentes de replicação usando o Assistente de Instalação do SQL Server ou um prompt de comando. Se você instalar componentes de replicação ao modificar uma instância existente de SQL Server, é necessário parar e reiniciar o agente de SQL Server quando a instalação for concluída. Essa ação ajuda a assegurar que o SQL Server Agent reconheça os subsistemas de agente de replicação e possa chamar agentes de replicação em etapas de trabalho.

Gabarito: C

11. Ano: 2015 Banca: CESPE Órgão: MEC Prova: TÉCNICO DE NÍVEL SUPERIOR - ANALISTA DE SISTEMAS

Julgue os itens subsequentes, relativos ao Microsoft SQL Server.

[1] SQL Server fornece um conjunto de tipos de dados primitivos tipos de cadeia de strings de tamanho fixo e variável até 2^{90} .

Comentário: Primeiramente 2^{90} (dois elevado a 90) é um número gigantesco para ser usado como tamanho de String. Em segundo lugar, o SQL SERVER passa como parâmetros n (char(n) e varchar(n) a quantidade de bytes e não a quantidade de caracteres como o PostgreSQL e o MySQL. Esse valor é limitado a 8000 bytes.

Gabarito: E

12. Ano: 2015 Banca: CESPE Órgão: MEC Prova: TÉCNICO DE NÍVEL SUPERIOR - ANALISTA DE SISTEMAS

Julgue os itens subsequentes, relativos ao Microsoft SQL Server.

[1] VIEW é uma tabela virtual cujo conteúdo está definido por uma instrução SELECT.

Comentário: A cláusula SELECT dentro do comando de criação da visão vai definir o conteúdo que será exibido. A instrução pode usar mais de uma tabela e outras visões. Permissões apropriadas são necessárias para selecionar os objetos referenciados na cláusula SELECT.

Uma view não tem que ser um subconjunto simples das linhas e colunas de uma determinada tabela. Ela pode ser criada usando mais de uma tabela ou outras exibições com uma cláusula VIEW de qualquer complexidade.

Gabarito: C.

13. ANO: 2014 BANCA: CESPE ÓRGÃO: ANATEL PROVA: ANALISTA ADMINISTRATIVO - SUPORTE E INFRAESTRUTURA DE TI

A respeito de banco de dados, julgue os itens que se seguem.



[1] É válida para o PostgreSQL 9.3, mas não para o SQL Server 2012, a criação da SEQUENCE seja por meio do seguinte comando:

```
CREATE SEQUENCE seqa START WITH 1;
```

Comentário: O comando acima, quando executado no SQL Server cria uma SEQUENCE. Ou seja, cria um objeto de sequência e especifica suas propriedades. Uma sequência é um objeto associado a um esquema definido pelo usuário que gera uma sequência de valores numéricos de acordo com a especificação com a qual a sequência foi criada.

A sequência de valores numéricos é gerada em ordem crescente ou decrescente em um intervalo definido e pode ser configurada para reiniciar (em um ciclo) quando se esgotar. As sequências, ao contrário de colunas de identidade, não são associadas a tabelas específicas. Os aplicativos fazem referência a um objeto de sequência para recuperar seu próximo valor. A relação entre sequências e tabelas é controlada pelo aplicativo. Os aplicativos de usuário podem referenciar um objeto de sequência e coordenar os valores nas várias linhas e tabelas.

Ou seja, o comando é válido tanto para o SQL Server quanto para o PostgreSQL.

Gabarito: C

14. ANO: 2014 BANCA: CESPE ÓRGÃO: ANTAQ PROVA: ANALISTA ADMINISTRATIVO - ANALISTA DE INFRAESTRUTURA

Acerca do Microsoft SQL Server 2008, julgue os seguintes itens.

[1] Para desabilitar uma trigger DDL (data definition language) definida com escopo de servidor (on all server), é necessária a permissão control server no servidor.

Comentário: Para desabilitar um gatilho DDL definido com escopo de servidor (ON ALL SERVER) ou um gatilho de logon, um usuário deve ter a permissão CONTROL SERVER no servidor. Para desabilitar um gatilho DDL definido com escopo de banco de dados (ON DATABASE), no mínimo, um usuário deve ter a permissão ALTER ANY DATABASE DDL TRIGGER no banco de dados atual. Logo, temos uma alternativa correta.

Gabarito: C

15. ANO: 2014 BANCA: CESPE ÓRGÃO: ANTAQ PROVA: ANALISTA ADMINISTRATIVO - ANALISTA DE INFRAESTRUTURA

Acerca do Microsoft SQL Server 2008, julgue os seguintes itens.

[1] O argumento clustered do comando create index cria um índice em que a ordem lógica dos valores da chave determina a ordem física das linhas correspondentes em uma tabela.

Comentário: O argumento CLUSTERED cria um índice no qual a ordem lógica dos valores de chave determina a ordem física das linhas correspondentes em uma tabela. O nível inferior, ou folha, do índice clusterizado contém as linhas de dados reais da tabela. Uma tabela ou visão pode ter apenas um índice clusterizado por vez.

Gabarito: C.

16. ANO: 2014 BANCA: CESPE ÓRGÃO: TC-DF PROVA: ANALISTA DE ADMINISTRAÇÃO



PÚBLICA - SISTEMAS DE TECNOLOGIA DA INFORMAÇÃO

Julgue os seguintes itens, acerca de sistemas de gerenciamento de bancos de dados (SGBD) e de cópias de segurança de dados.

[1] No SQL Server 2012, a criação de índices em tabelas temporárias locais pode ser feito off-line, desde que a tabela não possua tipos de dados LOB (Large Object).

Comentário: Se a tabela possuir LOB ela precisará ser criada off-line, quando você executar operações de índice online, as diretrizes seguintes se aplicam:

Os índices clusterizados devem ser criados, recriados ou descartados off-line quando a tabela subjacente contiver estes tipos de dados LOB (objeto grande): image, ntext e text.

Índices não clusterizados não exclusivos podem ser criados online, quando a tabela contiver tipos de dados LOB, mas nenhuma dessas colunas são usadas na definição de índice seja como colunas-chaves ou colunas não chave.

Os índices em tabelas temporárias locais, não podem ser criados, recriados ou soltos offline. Esta restrição não se aplica a índices em tabelas temporárias globais.

Índices podem ser retomados de onde pararam após uma falha inesperada, failover de banco de dados ou um comando PAUSE.

Gabarito: E.

17. CEBRASPE (CESPE) - Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

No Sql Server, um becape

- a) de arquivo agrega um ou mais arquivos ou grupos de arquivos de banco de dados.
- b) parcial agrega dados que ainda não foram afetados por COMMIT, tanto de dados quanto de transações.
- c) completo de banco de dados agrega todos os dados de todos os bancos de dados no momento em que o becape é concluído, com exceção dos logs de transação.
- d) diferencial agrega apenas logs de transações, incluindo somente transações feitas desde seu último becape de log até a transação mais recente.
- e) de logs de transações inclui todos os registros de log de forma cumulativa, independentemente de ter havido becape de log anterior ou becape completo.

Comentário: Vamos comentar cada uma das alternativas acima. Segundo a própria documentação do Microsoft SQL Server:

- a) **Correto.** Backup de arquivo é um backup de um ou mais arquivos ou grupos de arquivos de banco de dados.
- b) **Errado.** Backup parcial contém dados apenas de alguns grupos de arquivos em um banco de dados, incluindo os dados no grupo de arquivos primário, em cada grupo de arquivos de leitura/gravação e em qualquer arquivo somente leitura especificado opcionalmente. Contudo, o backup é feito com dados completamente escritos, ou seja, afetados por COMMIT.



- c) **Errado.** Backup completo contém todos os dados em um banco de dados ou em um conjunto de grupos de arquivos ou arquivos. Além disso, contém log suficiente para permitir a recuperação desses dados.
- d) **Errado.** Backup diferencial se baseia no backup completo mais recente de um banco de dados completo ou parcial ou um conjunto de arquivos de dados ou grupos de arquivos (a base diferencial) que contém somente as extensões de dados alterados desde a base diferencial.
- e) **Errado.** Backup de logs de transações que inclui todos os registros de log dos quais não foi feito backup em um backup de log anterior. Ou seja, não existe essa acumulação independente de ter havido backup de log anterior ou backup completo.

Além desses tipos, existem outros, tal como o backup: somente cópia (copy-only) - de uso especial que é independente da sequência regular dos backups do SQL Server.

Gabarito: A.

18. CEBRASPE (CESPE) - Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

Acerca das soluções presentes no Sql Server 2008 R2, assinale a opção correta.

- a) Reporting services fornece uma plataforma de comunicação embasada em mensagens assíncronas que permite a interoperabilidade entre sistemas.
- b) Integration services permite aos usuários integrar e gerenciar estruturas multidimensionais que contenham dados agregados de outras fontes de dados, como bancos de dados relacionais.
- c) SharePoint services é uma solução utilizada para construir soluções de integração de dados, incluindo a extração, transformação e carregamento de dados.
- d) Master data services visa integrar sistemas analíticos e operacionais distintos de modo que seja criada e gerenciada uma fonte de informações central e auditável.
- e) Service broker é uma ferramenta voltada para a criação de cluster de alta disponibilidade, permitindo que os dados sejam acessados e recuperados de forma distribuída, e, em caso de indisponibilidade em um dos nós da solução cluster, há recuperação automática dos dados.

Comentário: Perceba que as definições das alternativas incorretas se referem a outras ferramentas do SQL Server. Conforme a documentação do *Microsoft SQL Server e Sharepoint*:

- a) **Errado.** Reporting Services é uma plataforma de relatório baseada em servidor que fornece funcionalidade de relatório abrangente para várias fontes de dados. Ademais, inclui um conjunto completo de ferramentas para criação, gerenciamento e entrega de relatórios e APIs que permitem aos desenvolvedores integrar ou estender dados e processamento de relatório em aplicativos personalizados.
- b) **Errado.** Integration Services é uma plataforma para criar integração de dados em nível corporativo e soluções de transformações de dados. Ele inclui um conjunto completo de tarefas e transformações internas, ferramentas para construção de pacotes e o serviço para execução e gerenciamento de pacotes.



c) **Errado.** *SharePoint Services* serve como plataforma para aplicativos de servidor, como o *Microsoft Office SharePoint Server 2007*, a fim ajudar as organizações, equipes e unidades de negócios para serem mais eficazes, conectando pessoas e informações.

d) **Correto.** *Master Data Services* é a solução do *SQL Server* para gerenciamento de dados mestre (MDM), que descreve os esforços de uma organização para descobrir e definir listas não transacionais de dados, visando compilar listas mestre sustentáveis. *Master Data Services* inclui outros recursos, como hierarquias, segurança granular, transações, controle de versão de dados e regras de negócios.

e) **Errado.** *Service Broker* oferece suporte nativo para aplicativos de mensagens e enfileiramento no Mecanismo de Banco de Dados do *SQL Server*. Além disso, possibilita aos desenvolvedores criarem facilmente aplicativos *distribuídos e confiáveis*.

Gabarito: D.

19. CEBRASPE (CESPE) - Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

Acerca de tuning de banco de dados, bem como de técnicas de análise de desempenho e otimização de consultas em *SQL Server 2008 R2*, assinale a opção correta.

- a) Se, no plano de execução de determinada consulta, o DBA visualizou um ícone na execução da consulta representando a operação *bitmap*, então a consulta encontrou, em determinado momento, um campo do tipo *BLOB* (binary large object) ou um índice do tipo *fulltext*.
- b) A opção *fillfactor* no índice com valor 0 indica que não haverá folheamento de índices, o que se traduz em velocidade, ao passo que, no índice com valor 100, essa opção indica que não haverá *splits*, porém maior necessidade de espaço para armazenamento de dados.
- c) Se o *Sql Server* estiver instalado em uma máquina com mais de um processador, ele pode decidir a quantidade de processadores a serem utilizados no processamento da consulta, cujo valor é limitado ao número máximo de processadores conforme configurado na opção *max degree of parallelism*. A opção *MAXDOP* na criação de um índice pode mudar essa configuração durante a execução de tal índice.
- d) Uma tabela que possua uma chave primária composta exige que seus dados sejam fisicamente classificados e armazenados com base em seus valores de chave e, desse modo, por padrão, o *Sql Server* cria um índice clusterizado (*CLUSTERED*) na tabela para cada atributo pertencente à chave primária.
- e) Se, no plano de execução de determinada consulta, o DBA visualizou um ícone na execução da consulta representando a operação *the nested loops*, então a consulta utilizou, em determinado momento, um indicador de fila ou chave de *clustering* para pesquisar a linha correspondente na tabela ou índice clusterizado.

Comentário: Vamos analisar cada uma das alternativa ...

a) **Errado.** A presença do ícone *bitmap* não indica que a consulta encontrou um campo do tipo *BLOB* ou índice do tipo *fulltext*. O *SQL Server* usa o operador *Bitmap*, que é um operador



físico, para implementar filtro de bitmap em planos de consulta paralelos. Esse filtro de *bitmap* acelera a execução de consulta eliminando linhas com valores de chave que não podem produzir nenhum relatório de junção antes de passar as linhas por outro operador, como o operador *Parallelism*.

b) **Errado.** A opção *fillfactor* (fator de preenchimento) é fornecida para ajustar o armazenamento e o desempenho de dados de índice. Ao se criar ou recriar um índice, o valor desse fator determina a porcentagem de espaço em cada página de nível de folha a ser preenchida com dados, reservando o restante como espaço livre para futuro crescimento. O valor do *fillfactor* é uma porcentagem de 1 a 100 e o padrão para todo o servidor é zero (0), o que significa que as páginas de nível de folha estão totalmente preenchidas.

c) **Correto.** Exatamente! Quando o SQL Server é executado em um computador com mais de um microprocessador ou CPU, ele detecta o melhor grau de paralelismo, ou seja, o número de processadores utilizados para executar uma única instrução, para cada execução paralela de plano. É possível usar a opção *max degree of parallelism* para limitar o número de processadores a serem usados na execução paralela do plano. E o *MAXDOP* pode realmente substituir o valor de *max degree of parallelism*.

d) **Errado.** O índice clusterizado, além de melhorar o desempenho da consulta, pode ser recompilado ou reorganizado sob demanda para controlar a fragmentação da tabela. A criação de índice clusterizado não está restrito à ocorrência de chave primária composta. Ao ser criada uma restrição *PRIMARY KEY*, um índice clusterizado exclusivo é criado automaticamente, se um índice clusterizado ainda não existir na tabela e não for especificado um índice não clusterizado exclusivo.

e) **Errado.** Na verdade, o operador *Nested Loops*, que é um operador físico, executa operações lógicas de *inner join*, *left outer join*, *left semi join* e *left anti semi join*.

Gabarito: C.

20.ANO: 2013 BANCA: CESPE ÓRGÃO: ANTT PROVA: ANALISTA
ADMINISTRATIVO - INFRAESTRUTURA DE TI

A respeito de SQL Server, julgue os itens subsecutivos.

[1] Um dos recursos da LINQ (language integrated query) disponíveis no SQL Server 2008 é a realização de consultas, pelo desenvolvedor, diretamente em base de dados via framework.

Comentário: Power View oferece relatórios ad hoc intuitivos para usuários comerciais, tais como analistas de dados, tomadores de decisões de negócios e operadores de informações. Eles podem criar e interagir facilmente com exibição de dados de modelos tabulares com base em pastas de trabalho Power Pivot publicadas em uma Galeria do Power Pivot ou modelos de tabela criados usando o SQL Server Data Tools (SSDT) e, em seguida, implantados em instâncias do SQL Server 2017 Analysis Services. Power View é um aplicativo do Silverlight baseado no navegador, iniciado no SharePoint Server 2010 ou posterior.

Ao criar projetos de modelo tabular no SQL Server Data Tools, você pode configurar determinadas propriedades de relatório exclusivas para relatórios do Power View. Assim, o gabarito é certo.



Gabarito: C.

21. CESPE - Analista Administrativo (TCE-ES)/Informática/2013

A ferramenta presente no SQL Server que, após analisar uma carga de trabalho, pode recomendar a adição, remoção ou modificação de estruturas de design físicas em bancos de dados é denominada

- a) SQL Server Studio.
- b) SQL Server Admin.
- c) SQL Server Configuration Manager.
- d) Orientador de Otimização do Mecanismo de Banco de Dados.
- e) SQL Server Profiler.

Comentário: O DTA (Orientador de Otimização do Mecanismo de Banco de Dados) do Microsoft analisa bancos de dados e faz recomendações que você pode usar para otimizar desempenho de consulta. Você pode usar o Orientador de Otimização do Mecanismo de Banco de Dados para selecionar e criar um conjunto ideal de índices, exibições indexadas e partições de tabela sem precisar de conhecimentos avançados sobre a estrutura do banco de dados ou dos recursos internos do SQL Server. Com o DTA, é possível executar as tarefas a seguir.

Solucionar problemas de desempenho de uma consulta de problema específica
Ajustar um conjunto grande de consultas por um ou mais bancos de dados
Executar uma análise E-Se exploratória de possíveis alterações de design físico
Gerenciar o espaço de armazenamento

Portanto, o gabarito é a letra d).

Gabarito: D

22. CEBRASPE (CESPE) - Analista do Ministério Público da União/Tecnologia da Informação e Comunicação/Suporte e Infraestrutura/2013

Julgue o próximo item, acerca dos sistemas ORACLE, MySQL e SQL Server.

Na instalação do SQL Server 2012 em um sistema de arquivos NTFS, são definidas automaticamente as ACL (access control lists) em chaves de registro e em arquivos, devendo as ACL ser revisadas imediatamente após a conclusão da instalação.

Comentário: Durante a instalação, o SQL Server definirá ACLs apropriadas em chaves do Registro e arquivos se ele detectar NTFS. Essas permissões não devem ser alteradas. Logo, temos uma alternativa errada.

Gabarito: E.

23. ANO: 2013 BANCA: CESPE ÓRGÃO: STF PROVA: ANALISTA JUDICIÁRIO - SUPORTE EM TECNOLOGIA DA INFORMAÇÃO



A respeito de configuração e administração de bancos de dados, julgue os itens a seguir.

[1] No processo de instalação do SQL Server 2012, deve-se instalar apenas uma cópia das ferramentas de gerenciamento, independentemente da quantidade de instâncias do SQL Server instaladas na máquina.

Comentário: Independentemente do número de instâncias do SQL Server, do Analysis Services ou do Reporting Services instaladas no computador, será instalada apenas uma cópia das Ferramentas de Gerenciamento do SQL Server.

Gabarito: C.

24. ANO: 2013 BANCA: CESPE ÓRGÃO: STF PROVA: ANALISTA JUDICIÁRIO - SUPORTE EM TECNOLOGIA DA INFORMAÇÃO

Julgue os itens subsecutivos, com relação ao tuning de banco de dados.

[1] No SQL Server, o uso de variáveis de tabela permite aumentar o desempenho de determinadas consultas.

Comentário: Variável tipo TABLE nada mais é do que um tipo especial de variável que pode ser utilizada para armazenamento temporário de dados, de maneira similar a tabelas temporárias, não tem relação com o aumento de desempenho.

Gabarito: E.

25. ANO: 2013 BANCA: CESPE ÓRGÃO: CRPM PROVA: ANALISTA EM GEOCIÊNCIAS - SISTEMAS DE INFORMAÇÃO

Com relação a técnicas de análise de desempenho e otimização de consultas SQL, julgue os itens seguintes.

[1] No SQL Server, o comando TRUNCATE TABLE remove todos os dados de uma tabela e, se a tabela tiver uma coluna de identidade, provoca a reinicialização do contador de identidade.

Comentário: O truncate remove todas as linhas de uma tabela ou partições especificadas de uma tabela sem registrar as exclusões de linha individual. O TRUNCATE TABLE é semelhante à instrução DELETE sem nenhuma cláusula WHERE; entretanto, TRUNCATE TABLE é mais rápida e utiliza menos recursos de sistema e log de transações. No SQL Server, uma operação TRUNCATE TABLE pode ser revertida.

Se a tabela contiver uma coluna de identidade, o contador daquela coluna será redefinido no valor da semente definido para a coluna. Se não for definida nenhuma semente, o valor padrão utilizado será 1. Para manter o contador de identidade, use DELETE.

Gabarito: C.

26. ANO: 2013 BANCA: CESPE ÓRGÃO: TCE-RO PROVA: AUDITOR DE CONTROLE EXTERNO - CIÊNCIAS DA COMPUTAÇÃO



Julgue os itens subsequentes, com base nos conceitos de modelagem relacional dos dados e de administração de dados.

[1] Os becares gerados por uma versão mais recente do SQL Server não podem ser restaurados com o uso de versões anteriores.

Comentário: Nenhum backup do SQL Server pode ser restaurado para uma versão anterior do SQL Server a não ser na versão na qual o backup foi criado. Backups do master, model e msdb que foram criados em uma versão anterior do SQL Server não podem ser restaurados pelo SQL Server 2019 (15.x).

Cada versão do SQL Server usa um caminho padrão diferente das versões anteriores. Assim, para restaurar um banco de dados que foi criado no local padrão dos backups de versões anteriores, você deve usar a opção MOVE.

Gabarito: C.



27. ANO: 2013 BANCA: CESPE ÓRGÃO: BACEN PROVA: ANALISTA DO BACEN - ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Considerando os sistemas de gerenciamento de bancos de dados, julgue o próximo item.

[1] PostgreSQL, MariaDB, IIS e SQL Server são exemplos de sistemas gerenciadores de bancos de dados.

Comentário: MariaDB é um servidor de banco de dados que oferece a funcionalidade e substituição para o MySQL. MariaDB é construído por alguns dos autores originais do MySQL, com a ajuda da grande comunidade de desenvolvedores de software livre e software de código aberto. Além das funcionalidades básicas do MySQL, MariaDB oferece um rico conjunto de aprimoramentos de recursos, incluindo mecanismos de armazenamento alternativo, otimizações de servidores e patches. Então a questão está certa? Por que você deve saber que PostgreSQL e SQL Server são SGBDs ... Mas e IIS? O que danado é isso? É servidor de aplicações web da Microsoft. Logo, a alternativa está errada.

Gabarito: E

28. ANO: 2012 BANCA: CESPE ÓRGÃO: TJ-AC PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SISTEMAS

Com relação a banco de dados, julgue os itens que se seguem.

[1] No SQL SERVER 2012, permite-se a combinação da cláusula TOP com OFFSET e FETCH no mesmo escopo de consulta.

Comentário: Use OFFSET e FETCH na cláusula ORDER BY em vez da cláusula TOP para implementar uma solução de consulta com paginação. Uma solução de paginação (ou seja, o envio de pedaços ou "páginas" de dados para o cliente) é mais fácil de implementar usando as cláusulas OFFSET e FETCH. A sintaxe do comando não permite a utilização de ambos em uma mesma consulta.

Gabarito: E.

29. CEBRASPE (CESPE) - Analista Judiciário (TJ AC)/Técnico-Administrativa/Analista de Sistemas/2012

Com relação a banco de dados, julgue o item que se segue.

Quando a instrução SELECT DISTINCT é utilizada no SQL SERVER 2012, os nomes e aliases de coluna deverão ser definidos na lista de seleção.

Comentário: Essa é uma questão padrão que trata de conceitos globais de SQL. A lista de colunas e alias devem ser definidos na lista da seleção.

Gabarito: C.

30. ANO: 2012 BANCA: CESPE ÓRGÃO: TJ-AC PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SUPORTE



A respeito do banco de dados SQL Server e da linguagem SQL, julgue os itens seguintes.

[1] O comando SELECT GETDATE() recebe como retorno a data e a hora atual do sistema operacional. Para se retornar um usuário qualquer conectado a uma base de dados, deve-se executar o comando SELECT SYSTEM_ADMIN.

Comentário: O GETDATE() está descrito corretamente. Já o SYSTEM_ADMIN não! O certo seria utilizar um desses comandos: SYSTEM_USER() e CURRENT_USER(). SYSTEM_USER permite que um valor fornecido pelo sistema para o logon atual seja inserido em uma tabela quando nenhum valor padrão é especificado. Já o CURRENT_USER é uma função que retorna o nome do usuário atual; essa função é equivalente a USER_NAME(). Vejamos um exemplo da utilização destes comandos na instrução de CREATE TABLE para fornecer um valor DEFAULT:

```
CREATE TABLE Vendas.Rastreio_Vendas
(
  Territorio_id int IDENTITY(2000, 1) NOT NULL,
  Rep_id int NOT NULL,
  Ultima_venda datetime NOT NULL DEFAULT GETDATE(),
  SRep_rastreio_user varchar(30) NOT NULL DEFAULT SYSTEM_USER
);
```

Gabarito: E.

31. ANO: 2012 BANCA: CESPE ÓRGÃO: TJ-AC PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SUPORTE

A respeito do banco de dados SQL Server e da linguagem SQL, julgue os itens seguintes.

[1] Ao se executar o comando SELECT ROUND(2.1234,1) AS Valor1, obtém-se Valor1 2.1000 como resultado.

Comentário: Retorna um valor numérico, arredondado, para o comprimento ou precisão especificados.

Gabarito: C.



QUESTÕES COMENTADAS - MULTIBANCAS

1. ISS São José dos Campos/FGV/2024

Com relação aos spinlock no MS SQL Server 2019, avalie as afirmativas a seguir.

- I. Os spinlocks não são uma exclusividade do MS SQL Server. São utilizados pelo sistema operacional quando é necessário obter acesso a uma determinada estrutura de dados somente por um curto intervalo de tempo. Por exemplo, quando uma thread não consegue obter acesso ao tentar adquirir um spinlock, ela é executada de maneira periódica se o recurso está disponível ou não, em vez de fazer uma suspensão imediata. Após algum tempo, a thread que está aguardando o spinlock será suspenso antes de poder adquirir o recurso. A suspensão permite que outras threads em execução na mesma CPU sejam executadas.
- II. O MS SQL Server usa spinlocks para proteger o acesso a algumas estruturas internas de dados. Os spinlocks são usados no mecanismo para serializar o acesso a determinadas estruturas de dados de maneira semelhante aos locks.
- III. A principal diferença entre o lock e o spinlock é o fato de que os locks executar um loop por um período verificando a disponibilidade de uma estrutura de dados. Por outro lado, uma thread tentando obter acesso a uma estrutura protegida por spinlock cederá imediatamente se recurso não estiver disponível. A suspensão requer a alternância de contexto de uma thread fora da CPU para que outra thread possa ser executada.

Está correto o que se afirma em

- a) III, apenas.
- b) II e III.
- c) II, apenas.
- d) I e II.
- e) I e III.

Comentário: As afirmativas estão misturadas entre conceitos corretos e incorretos sobre os spinlocks no MS SQL Server 2019. Vamos avaliá-las individualmente:

- I. Esta afirmativa está correta. Spinlocks são exclusivos do MS SQL Server, e também são usados pelo sistema operacional para esse propósito.
- II. Esta afirmativa está correta. O MS SQL Server usa spinlocks para proteger o acesso a algumas estruturas internas de dados, como parte de seu mecanismo de controle de concorrência.
- III. Esta afirmativa está incorreta. A principal diferença entre lock e spinlock não está relacionada à disponibilidade imediata da estrutura de dados, mas sim ao comportamento quando a estrutura não está disponível. Locks geralmente resultam em suspensão da thread até que a estrutura de dados esteja disponível, enquanto spinlocks executam um loop ocupado, verificando continuamente a disponibilidade do recurso.

Portanto, a resposta correta é a alternativa d) I e II, apenas.

Complementando: um spinlock é um bloqueio que faz com que um thread que tenta adquiri-lo simplesmente espere em um loop ("spin") enquanto verifica repetidamente se o bloqueio está disponível. Como o thread permanece ativo, mas não executa uma tarefa útil, o uso de tal bloqueio é uma espécie de espera ocupada. Uma vez adquiridos, os spinlocks geralmente serão mantidos até



2. Câmara Municipal de São Paulo/FGV/2024

Com relação ao Microsoft SQL Server Profiler, avalie se as afirmativas a seguir são falsas (F) ou verdadeiras (V).

- I. É uma interface gráfica do usuário para o SQL Trace para monitorar uma instância do SGBD ou do Analysis Services. Permite que o analista monitore um ambiente de produção para ver quais procedimentos armazenados estão afetando o desempenho por serem executados com muita lentidão.
- II. Permite que o analista capture uma série de instruções Transact-SQL que levam a um problema de desempenho. O rastreamento salvo pode então ser usado para replicar o problema em um servidor de teste onde o problema pode ser diagnosticado.
- III. Oferece suporte à auditoria das ações executadas em instâncias do SGBD SQL Server. As auditorias registram ações relacionadas à segurança para revisão posterior por um administrador de segurança.

As afirmativas são, respectivamente,

- a) V–V–V.
- b) F–F–V.
- c) V–F–F.
- d) F–V–V.
- e) F–F–F.

Comentário: A análise correta das afirmativas é: a) V–V–V.

I. Verdadeiro. O Microsoft SQL Server Profiler é uma ferramenta gráfica que permite monitorar uma instância do SQL Server ou do Analysis Services. Ele pode ser usado para identificar procedimentos armazenados ou outras consultas que estão afetando o desempenho do sistema.

II. Verdadeiro. O Profiler permite capturar uma série de instruções Transact-SQL que levam a problemas de desempenho. Os rastreamentos podem ser salvos e reproduzidos em um servidor de teste para diagnóstico posterior.

III. Verdadeiro. O SQL Server Profiler oferece suporte à auditoria das ações executadas no SQL Server. Isso inclui a capacidade de registrar ações relacionadas à segurança para revisão posterior por um administrador de segurança.

3. Câmara Municipal de São Paulo/FGV/2024

Os processos de otimização de consultas SQL em ambientes Oracle podem utilizar diversos tipos de



O tipo de índice adequado para colunas que possuem um número relativamente baixo de valores, para aplicações de armazenamento de dados em que há baixa atividade de declarações SQL do tipo DML e filtragens ad hoc e eficiente em consultas com a função COUNT () denomina-se índice

- a) baseado em função.
- b) chave reversa.
- c) particionado.
- d) árvore B.
- e) bitmap.

Comentário: O tipo de índice adequado para colunas que possuem um número relativamente baixo de valores e são eficientes em consultas com a função COUNT() é o índice: e) bitmap.

Os índices bitmap são especialmente úteis para colunas que possuem um número limitado de valores distintos, o que é comum em cenários de baixa cardinalidade. Eles são eficientes para consultas que envolvem operações como COUNT(), pois permitem que o banco de dados faça uma varredura eficiente nos bits correspondentes aos valores de interesse. Isso pode resultar em um desempenho melhor do que outros tipos de índices em certas situações, especialmente quando há baixa atividade de DML e filtragens ad hoc.

Gabarito: E

4. FUNDATEC - Ana Sist (BRDE)/BRDE/Administração de Banco de Dados/2023

O DBA de um banco de dados SQL Server 2019 precisou procurar no catálogo algumas definições existentes sobre um esquema da base de dados específico para executar o seu trabalho. Ele descobriu que:

- Para procurar as tabelas existentes, deveria consultar a SYS.TABLES, e para encontrar as chaves primárias existentes, poderia consultar a tabela _____;
- Encontraria na tabela _____ o identificador do tipo de dado associado aos atributos de uma dada tabela;
- Para encontrar os procedimentos armazenados na base de dados, precisava consultar a SYS.PROCEDURES, e que se quisesse saber detalhes dos parâmetros de um dado procedimento armazenado, poderia encontrá-los na _____;

Assinale a alternativa que preenche, correta e respectivamente, as lacunas do trecho acima.

- a) SYS.INDEXES – SYS.COLUMNS – SYS.PARAMETERS
- b) SYS. INDEXES _KEYS – SYS.TYPES – SYS.PARAMETERS
- c) SYS.INDEXES – SYS. COLUMNS – SYS.PROCEDURES_EXTENDED
- d) SYS.INDEXES – SYS.TYPES – SYS.PARAMETERS



Comentário: O gabarito correto é de fato a letra A: a) SYS.INDEXES – SYS.COLUMNS – SYS.PARAMETERS.

No SQL Server 2019, o DBA (Administrador de Banco de Dados) precisa consultar o catálogo do sistema para obter informações sobre as definições do esquema do banco de dados. O catálogo do sistema é uma coleção de visualizações e funções que fornecem metadados sobre os objetos do banco de dados, como tabelas, colunas, procedimentos armazenados, chaves primárias, entre outros.

SYS.TABLES: Esta visualização contém uma linha para cada tabela do banco de dados. Ao consultar a SYS.TABLES, o DBA pode encontrar informações sobre as tabelas existentes, como seus nomes e esquemas.

SYS.INDEXES: Esta visualização contém uma linha para cada índice definido em uma tabela. Para encontrar as chaves primárias existentes, o DBA deve consultar a SYS.INDEXES e procurar pelos índices que são marcados como chaves primárias.

SYS.COLUMNS: Esta visualização contém uma linha para cada coluna em cada tabela do banco de dados. Se o DBA quiser encontrar o identificador do tipo de dados associado aos atributos de uma tabela, ele pode consultar a SYS.COLUMNS para obter informações sobre os tipos de dados das colunas.

SYS.PARAMETERS: Esta visualização contém uma linha para cada parâmetro de cada procedimento armazenado no banco de dados. Para encontrar detalhes dos parâmetros de um procedimento armazenado, o DBA pode consultar a SYS.PARAMETERS e filtrar os resultados para o procedimento de interesse.

Portanto, ao consultar estas visualizações do catálogo do sistema, o DBA pode obter informações detalhadas sobre as tabelas, colunas, índices e procedimentos armazenados em um banco de dados SQL Server 2019, o que é essencial para executar suas tarefas de administração e manutenção do banco de dados.

Gabarito: A

5. FUNDATEC - Ana Sist (BRDE)/BRDE/Administração de Banco de Dados/2023

Considere que o DBA de um banco de dados SQL Server 2019 definiu o índice abaixo sobre a tabela TAB1:

```
CREATE NONCLUSTERED INDEX IDX1  
ON TAB1 (codcomp, dataInicial)  
WHERE DataFinal IS NOT NULL;
```



I. Esse comando cria um índice filtrado.

II. O índice criado por esse comando contribui à melhoria do desempenho de consultas principalmente quando a condição (*DataFinal IS NOT NULL*) é observada em um grande número de tuplas dessa tabela.

III. O formato de armazenamento primário do índice criado por esse comando é columnstore.

Quais estão corretas?

a) Apenas I.

b) Apenas II.

c) Apenas III.

d) Apenas II e III.

e) I, II e III.

Comentário: A assertiva I está correta, pois o comando `CREATE NONCLUSTERED INDEX` com a cláusula `WHERE DataFinal IS NOT NULL` cria um índice filtrado. Esse tipo de índice permite que você defina uma condição na qual apenas as linhas que atendem a essa condição são indexadas.

A assertiva II está incorreta porque o índice criado com a cláusula `"WHERE DataFinal IS NOT NULL"` não contribui diretamente para a melhoria do desempenho de consultas quando essa condição é aplicada em um grande número de tuplas da tabela TAB1.

A assertiva III está incorreta. O formato de armazenamento primário do índice criado por esse comando é `NONCLUSTERED`, não `columnstore`. O índice `columnstore` é uma tecnologia diferente usada para armazenar dados de forma otimizada para consultas analíticas e análise de dados em grandes conjuntos de dados, mas não está relacionado ao índice criado neste comando.

Portanto, a resposta correta é a letra a) Apenas I

Gabarito: A

6. FUNDATEC - Ana Sist (BRDE)/BRDE/Administração de Banco de Dados/2023

Considere as seguintes assertivas sobre recursos de backup no SQL Server 2019, assinalando V, se



- () Oferece diferentes modelos de recuperação, entre eles o total, diferencial e log de transações.
- () Um backup diferencial captura apenas as extensões dos dados alterados desde o último backup diferencial.
- () Se a base de dados foi criada com múltiplos arquivos de dados, é possível criar backups de arquivos individuais e restaurá-los individualmente.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- a) V – F – V.
b) V – V – F.
c) F – F – V.
d) F – V – V.
e) F – F – F.

Comentário: A ordem correta é: c) F – F – V.

A primeira assertiva é falsa. O SQL Server oferece diferentes modelos de recuperação de backup, incluindo backup total (full), backup diferencial e backup do log de transações.

A segunda assertiva é falsa. Um backup diferencial captura todas as alterações desde o último backup total, não apenas as extensões dos dados alterados.

A terceira assertiva é verdadeira. Se uma base de dados foi criada com múltiplos arquivos de dados, é possível criar backups de arquivos individuais e restaurá-los individualmente.

Portanto, a resposta correta é a letra C.

Gabarito: C

7. FUNDATEC - ANC (PROCERGS)/PROCERGS/Suporte/Bancos de Dados/2023

Analise as assertivas abaixo sobre o sistema de banco de dados Microsoft SQL Server 2019 e assinale a alternativa correta.



II. Os três tipos de JOIN são INNER, OUTER e CROSS.

III. Colunas usadas em uma condição de JOIN precisam ser do mesmo tipo.

- a) Todas estão corretas.
- b) Todas estão incorretas.
- c) Apenas I está correta.
- d) Apenas II está correta.
- e) Apenas III está correta.

Comentário: A alternativa correta é: c) Apenas I está correta.

I. Correta. Durante operações de JOIN, valores nulos não são considerados uma combinação, pois o SQL Server trata a comparação de valores nulos de maneira especial. Portanto, valores nulos não se juntarão a nenhum valor durante um JOIN.

II. Incorreta. Os três tipos de JOIN no SQL Server são INNER JOIN, OUTER JOIN (que inclui LEFT, RIGHT e FULL) e CROSS JOIN. Não existe um tipo de JOIN específico chamado "CROSS".

III. Incorreta. As colunas usadas em uma condição de JOIN não precisam necessariamente ser do mesmo tipo. O SQL Server fará uma conversão implícita dos tipos de dados, se necessário, para realizar a comparação durante o JOIN.

Portanto, apenas a assertiva I está correta.

Gabarito: C

8. FUNDATEC - ANC (PROCERGS)/PROCERGS/Suporte/Bancos de Dados/2023

Assinale a alternativa que apresenta um caso de uso que é mais adequado para uso do Accelerated Database Recovery (ADR) do Microsoft SQL Server 2019.

- a) Um sistema operando com transações de longa duração.
- b) Quando o banco de dados está operando com mirroring (espelhamento).
- c) Em um sistema que executa muitas atualizações pequenas, mas não agrupadas.



e) Em sistemas onde o espaço de armazenamento é mais importante do que disponibilidade.

Comentário: A alternativa correta é: a) Um sistema operando com transações de longa duração.

O Accelerated Database Recovery (ADR) é uma funcionalidade do Microsoft SQL Server 2019 projetada para melhorar o desempenho e a disponibilidade do banco de dados, especialmente em cenários com transações de longa duração. Ele ajuda a reduzir o tempo de recuperação e a minimizar os bloqueios durante operações de rollback e recuperação.

Portanto, o ADR é mais adequado para sistemas que operam com transações de longa duração, onde a recuperação rápida e eficiente é essencial para manter a integridade e a disponibilidade dos dados.

Gabarito: A

9. (VUNESP - ATCE (TCM SP)/TCM SP/Técnico de Informática/2023)

O comando do Transact-SQL para criar um *stored procedure*, denominado Eight, passando o parâmetro Seven, para executar uma operação de seleção é:

a) CREATE PROC @Eight #Seven AS

SELECT ...;

b) CREATE PROC &Eight \$Seven AS

SELECT ...;

c) CREATE PROC Seven.Eight AS

SELECT ...;

d) CREATE PROC #Seven OF Eight AS

SELECT ...;

e) CREATE PROC Eight @Seven AS

SELECT ...;

Gabarito: E

Comentário: A opção correta para criar um stored procedure em Transact-SQL, denominado Eight, passando o parâmetro Seven, para executar uma operação de seleção é: e) CREATE PROC Eight @Seven AS SELECT ...;



objetos que contêm instruções T-SQL reutilizáveis. Eles são armazenados no banco de dados e podem ser chamados e executados repetidamente.

A criação de um procedimento armazenado envolve o uso da declaração CREATE PROCEDURE. Aqui está um exemplo básico de criação de um procedimento armazenado simples:

```
CREATE PROCEDURE NomeDoProcedimento
```

```
    @Parametro1 TipoDado,
```

```
    @Parametro2 TipoDado
```

```
AS
```

```
BEGIN
```

```
    -- Lógica do procedimento aqui
```

```
    SELECT Coluna1, Coluna2
```

```
    FROM Tabela
```

```
    WHERE Condição = @Parametro1;
```

```
END;
```

Neste exemplo:

NomeDoProcedimento é o nome do procedimento armazenado.

@Parametro1 e @Parametro2 são parâmetros do procedimento, com tipos de dados específicos.

A seção AS BEGIN ... END contém a lógica do procedimento, que pode incluir instruções SQL, lógica condicional, loops, etc.

Os procedimentos armazenados são úteis para encapsular lógica de negócios complexa, melhorar a modularidade do código, permitir reutilização e facilitar a manutenção. Eles também oferecem benefícios de desempenho, pois o plano de execução pode ser armazenado em cache, resultando em execuções mais eficientes.

Para executar um procedimento armazenado, você pode usar a declaração EXEC ou EXECUTE:

```
EXEC NomeDoProcedimento @Parametro1 = Valor1, @Parametro2 = Valor2;
```

10. (VUNESP - ATCE (TCM SP)/TCM SP/Técnico de Informática/2023)



```
CREATE TRIGGER Yellow ON Red  
AFTER UPDATE  
AS RAISEERROR ('Aviso', 16, 10);
```

Esse comando cria um *trigger* denominado

- a) Yellow, que é acionado quando algum registro da tabela Red é alterado, gerando uma mensagem de aviso.
- b) Yellow, que é acionado quando algum registro do banco de dados Red é excluído, gerando uma interrupção com um aviso.
- c) Red, que é acionado quando algum registro da tabela Yellow é alterado, enviando um *e-mail* ao responsável.
- d) Red, que é acionado quando alguma tabela do banco de dados Yellow é modificado, enviando um *e-mail* ao responsável.
- e) Red, que é acionado quando alguma chave primária da tabela Yellow for alterada, provocando uma parada do banco de dados.

Gabarito: A

Comentário: O comando fornecido cria um trigger denominado Yellow que é acionado após uma operação de UPDATE na tabela Red, gerando uma mensagem de aviso com a função RAISEERROR. Portanto, a opção correta é a letra a.

11. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

O sistema gerenciador de bancos de dados Microsoft SQL Server 2012 possui a opção OFFSET que pode ser inserida em um comando SELECT. Tal opção especifica

- a) o número máximo de linhas que devem ser exibidas como resultado da execução do comando.
- b) o número de linhas a serem excluídas da tabela alvo, antes da exibição das linhas de resultado.
- c) o tempo mínimo para que a resposta solicitada seja exibida.
- d) o número máximo de usuários simultâneos no banco de dados em questão.
- e) quantas linhas devem ser ignoradas antes que o comando inicie a exibição das linhas de resultado.



Comentário: A opção OFFSET em um comando SELECT no Microsoft SQL Server especifica quantas linhas devem ser ignoradas antes que o comando comece a exibir as linhas de resultado. Portanto, a opção correta é: e) quantas linhas devem ser ignoradas antes que o comando inicie a exibição das linhas de resultado.

12. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

O comando do sistema gerenciador de bancos de dados Microsoft SQL Server que desfaz as transações até um *savepoint* especificado é:

- a) ROLLBACK TRAN @<nome do savepoint>;
- b) ROLLBACK TRAN <nome do savepoint>;
- c) ROLLBACK TRAN UNTIL <nome do savepoint>;
- d) ROLLBACK TRANSACTION FOR <nome do savepoint>;
- e) ROLLBACK TRANSACTION <nome do savepoint>;

Gabarito: Anulada

Comentário: O comando do sistema gerenciador de bancos de dados Microsoft SQL Server que desfaz as transações até um savepoint especificado é: b) ROLLBACK TRAN <nome do savepoint>; ou e) ROLLBACK TRANSACTION <nome do savepoint>;. Desta forma, à banca optou por anular a questão.

13. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

Considere o seguinte comando do sistema gerenciador de bancos de dados Microsoft SQL Server:

```
TRUNCATE TABLE Hand;
```

O resultado da execução desse comando é

- a) excluir metade das tabelas do banco de dados Hand.
- b) excluir todos os registros da tabela Hand.
- c) excluir metade dos registros da tabela Hand.
- d) limitar o tamanho de todos os registros da tabela Hand.
- e) limitar o número de tabelas do banco de dados Hand.



Comentário: O comando TRUNCATE TABLE Hand; no sistema gerenciador de bancos de dados Microsoft SQL Server remove todos os registros da tabela Hand sem afetar a estrutura da tabela. Portanto, a opção correta é: b) excluir todos os registros da tabela Hand.

14. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

A linguagem Transact SQL do sistema gerenciador de bancos de dados Microsoft SQL Server possui o comando BULK INSERT, cuja função é

- a) codificar os dados de uma tabela especificada nesse comando.
- b) confirmar as operações de inserção de dados realizadas em determinado período de tempo.
- c) importar um arquivo com dados para uma tabela ou visão da base de dados.
- d) restituir os registros excluídos de determinada tabela da base de dados.
- e) eliminar os registros que contenham valores nulos nas tabelas do banco de dados.

Gabarito: C

Comentário: A opção correta é: c) importar um arquivo com dados para uma tabela ou visão da base de dados.

O comando BULK INSERT no Transact-SQL (T-SQL) é utilizado para importar dados de um arquivo em massa para uma tabela ou visão em um banco de dados SQL Server. Ele permite carregar grandes volumes de dados de forma eficiente. Abaixo está a estrutura básica do comando BULK INSERT:

BULK INSERT

```
{ database_name.schema_name.table_name | schema_name.table_name | table_name }
```

```
FROM 'data_file'
```

```
WITH
```

```
(
```

```
[ [ , ] BATCHSIZE = batch_size ]
```

```
[ [ , ] CHECK_CONSTRAINTS ]
```

```
[ [ , ] CODEPAGE = { 'ACP' | 'OEM' | 'RAW' | 'code_page' } ]
```

```
[ [ , ] DATAFILETYPE =
```



```
[ [ , ] FIRSTROW = first_row ]  
[ [ , ] FIRE_TRIGGERS ]  
[ [ , ] FORMATFILE = 'format_file_path' ]  
[ [ , ] KEEPIDENTITY ]  
[ [ , ] KEEPNULLS ]  
[ [ , ] KILOBYTES_PER_BATCH = kilobytes_per_batch ]  
[ [ , ] LASTROW = last_row ]  
[ [ , ] MAXERRORS = max_errors ]  
[ [ , ] ORDER ( { column [ ASC | DESC ] } [ ,...n ] ) ]  
[ [ , ] ROWS_PER_BATCH = rows_per_batch ]  
[ [ , ] ROWTERM = 'row_terminator' ]  
[ [ , ] TABLOCK ]  
[ [ , ] CODEPAGE = { 'ACP' | 'OEM' | 'RAW' | 'code_page' } ]  
)
```

Alguns dos parâmetros mais comuns são:

FROM 'data_file': Especifica o arquivo que contém os dados a serem importados.

WITH: Indica o início da cláusula de opções.

FIELDTERMINATOR: Especifica o caractere que separa os campos no arquivo de dados.

ROWTERMINATOR: Especifica o caractere que marca o final de uma linha no arquivo de dados.

CHECK_CONSTRAINTS: Verifica as restrições de integridade referencial durante a operação.

Essa é uma estrutura básica e pode haver outras opções específicas dependendo dos requisitos do usuário e do formato do arquivo de dados. O comando BULK INSERT é uma ferramenta poderosa para carregar grandes volumes de dados de forma eficiente no SQL Server.

15. (VUNESP - Ana Sis (CM SBO)/CM SBO/2023)

O sistema gerenciador de banco de dados Microsoft SQL Server 2008 R2 permite replicar dados



- a) Parcial.
- b) Definida.
- c) Transacional.
- d) Permanente.
- e) Proporcional.

Gabarito: C

Comentário: O sistema gerenciador de banco de dados Microsoft SQL Server 2008 R2 oferece três tipos de replicação: a replicação parcial (Snapshot), a replicação definida (Transactional), e a replicação permanente (Merge).

Dessa forma, a resposta correta é: c) Transacional.

Lembrando que, os três principais tipos de replicação no Microsoft SQL Server são:

Snapshot Replication (Replicação de Instantâneo):

Neste tipo de replicação, uma cópia instantânea de dados é tirada em um determinado momento e propagada para os assinantes.

É adequado para situações em que a latência na entrega dos dados não é crítica e os dados não mudam com frequência.

Geralmente, é utilizado quando há uma necessidade de fornecer uma versão estática dos dados em intervalos regulares.

Transactional Replication (Replicação Transacional):

Este tipo de replicação é baseado em transações e é adequado para ambientes onde é crucial manter a consistência transacional entre o publicador e os assinantes.

As alterações no publicador são replicadas quase em tempo real para os assinantes, mantendo a integridade transacional.

É frequentemente utilizado em cenários onde há muitas atualizações frequentes nos dados.

Merge Replication (Replicação de Mesclagem):

A replicação de mesclagem é projetada para cenários em que as alterações podem ocorrer em qualquer lugar, seja no publicador, no assinante ou em ambos.

Ela permite que várias cópias dos dados existam em diferentes locais e, em seguida, as alterações são mescladas para garantir consistência.



É adequada para ambientes desconectados, onde os assinantes podem realizar atualizações offline e, posteriormente, mesclar as alterações.

Cada tipo de replicação tem seus casos de uso específicos, dependendo dos requisitos de consistência, latência e conectividade do ambiente de banco de dados.

16. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A unidade mais comum de medida de área é o hectare.

Gabarito: Errado

Comentário: O item apresenta uma informação incorreta. O hectare não é a unidade mais comum de medida de área; é uma unidade de medida de área, mas não é a mais comum. A unidade mais comum de medida de área depende do contexto geográfico e das práticas locais.

As unidades mais comuns de medida de área podem incluir metros quadrados (m²) e quilômetros quadrados (km²) em sistemas métricos, acres em alguns contextos anglo-americanos, entre outras unidades regionais.

Portanto, a afirmação de que o hectare é a unidade mais comum de medida de área é incorreta. A escolha da unidade de medida de área depende do sistema de unidades adotado e das práticas locais.

17. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A função ST_Area permite calcular a área de polígonos.

Gabarito: Certo

Comentário: A afirmação está correta. No SQL Server, a função ST_Area faz parte das funcionalidades relacionadas à extensão espacial (Spatial Extensions) e é usada para calcular a área de polígonos em um sistema de referência espacial (SRID) específico. A sintaxe geral da função ST_Area seria algo assim:

```
SELECT geometry_column.ST_Area() AS Area FROM your_table;
```

Essa consulta retornaria a área do polígono representado pela coluna geometry_column na tabela your_table. É importante observar que o resultado da área dependerá da unidade de medida do sistema de referência espacial utilizado. Portanto, se você precisa calcular a área de polígonos em um



18. Ano: 2018 Banca: FCC Órgão: DPE-AM Cargo: Analista Área: Banco de Sistemas Questão: 47.

Um Analista de Sistemas deseja alterar a coluna quantidade, que faz parte da tabela pedido do banco de dados empresa, do tipo INT para o tipo DECIMAL(7,2). Para isso, utilizando Transact-SQL no SQL Server, deverá usar o comando

- MODIFY COLUMN quantidade TO DECIMAL (7, 2) FROM empresa;
- ALTER TABLE empresa ALTER COLUMN quantidade DECIMAL (7, 2);
- ALTER TABLE empresa SET quantidade TO DECIMAL (7, 2);
- MODIFY TABLE empresa SET COLUMN quantidade DECIMAL (7, 2);
- ALTER TABLE empresa MODIFY COLUMN quantidade TO DECIMAL (7, 2);

Comentário: Sabemos que o comando ALTER TABLE que modifica uma definição de tabela alterando, adicionando ou eliminando **colunas e restrições**, reatribuindo e reconstruindo partições ou desativando ou ativando restrições e gatilhos. A sintaxe completa do comando pode ser encontrada [aqui](#)¹. Vejamos a sintaxe responsável pela alteração de colunas:

```
ALTER TABLE [ database_name . [ schema_name ] . | schema_name . ] table_name
{
    ALTER COLUMN column_name
    {
        [ type_schema_name. ] type_name
        [ (
            {
                precision [ , scale ]
                | max
                | xml_schema_collection
            }
        ) ]
        [ COLLATE collation_name ]
        [ NULL | NOT NULL ] [ SPARSE ]
        | { ADD | DROP }
        { ROWGUIDCOL | PERSISTED | NOT FOR REPLICATION | SPARSE | HIDDEN }
        | { ADD | DROP } MASKED [ WITH ( FUNCTION = ' mask_function ' ) ]
    }
    [ WITH ( ONLINE = ON | OFF ) ]
    | [ WITH { CHECK | NOCHECK } ]
}
```

Apenas pela sintaxe já podemos marcar a resposta na alternativa B. Outros pontos interessantes que podemos listar sobre a alteração de coluna. Um deles refere-se às restrições para que uma coluna possa ou não ser alterada. A coluna a ser modificada **não** pode ter nenhuma das seguintes características:

- Ser uma coluna com um tipo de dados *timestamp*.
- Ser o ROWGUIDCOL para a tabela.
- Ser uma coluna computada ou usada em uma coluna computada.
- Ser usada em estatísticas geradas pela instrução **CREATE STATISTICS**, a menos que (1) a coluna seja do tipo de dados *varchar*, *nvarchar* ou *varbinary*, o tipo de dados não seja alterado e o novo tamanho seja igual ou maior que o tamanho antigo ou (2) se a coluna for alterada de não nulo



- Ser usado em uma restrição PRIMARY KEY ou [FOREIGN KEY] REFERENCES.
- Ser usada em uma restrição CHECK ou UNIQUE. No entanto, a alteração do comprimento de uma coluna de tamanho variável usada em uma restrição CHECK ou UNIQUE é permitida.
- Ser associada a uma definição padrão. No entanto, o comprimento, a precisão ou a escala de uma coluna podem ser alterados se o tipo de dados não for alterado.

Gabarito: B

¹ <https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-table-transact-sql?view=sql-server-2017>



19. CESPE - Especialista Técnico (BNB)/Analista de Sistema/2018

Acerca de bancos de dados, julgue o item que se segue.

O código a seguir, criado no SQL Server 2017, apresenta uma visão materializada, especificamente devido ao argumento SCHEMABINDING.

```
CREATE VIEW VwTeste  
WITH SCHEMABINDING  
AS  
SELECT campo1 FROM tabela WHERE campo1 > 17;
```

Certo

Errado

Comentário: O comando CREATE VIEW (Transact-SQL) cria uma tabela virtual cujo conteúdo (colunas e linhas) é definido por uma consulta. Use esta instrução para criar uma exibição dos dados em uma ou mais tabelas no banco de dados. Segue sua sintaxe:

```
CREATE [ OR ALTER ] VIEW [ schema_name . ] view_name [ (column [ ,...n ] ) ]  
[ WITH <view_attribute> [ ,...n ] ]  
AS select_statement  
[ WITH CHECK OPTION ]  
[ ; ]  
  
<view_attribute> ::=  
{  
    [ ENCRYPTION ]  
    [ SCHEMABINDING ]  
    [ VIEW_METADATA ]  
}
```

O argumento SCHEMABINDING associa a exibição ao esquema da tabela ou tabelas subjacentes. Quando SCHEMABINDING for especificado, a tabela ou tabelas base não poderão ser modificadas de um modo que possam afetar a definição da exibição. A própria definição da exibição, primeiro, deve ser modificada ou descartada para remover as dependências na tabela a ser modificada.

Portanto, o comando não cria uma visão materializada. Assim, o gabarito é errado.

Gabarito: E

20. FCC - Agente de Fiscalização à Regulação de Transporte (ARTESP)/Tecnologia da Informação/2017

Em relação à remoção de linhas no SQL Server, é correto afirmar:

a) Uma tabela que tenha todas as linhas removidas permanece no banco de dados. A



- b) Ao utilizar uma instrução DELETE, se a cláusula WHERE não for especificada, apenas a primeira linha da tabela será excluída.
- c) Diferentemente da instrução DELETE, uma tabela esvaziada usando a instrução TRUNCATE TABLE é removida do banco de dados, junto com seus índices e outros objetos associados.
- d) Pode-se usar a cláusula UP (n) para limitar o número de linhas que são excluídas em uma instrução DELETE. Neste caso a operação de exclusão é executada em uma seleção aleatória de n linhas.
- e) Caso seja utilizada a cláusula UP junto com TRUNCATE para excluir linhas em uma determinada ordem, será preciso usar UP junto com ORDER BY em uma instrução de subseleção.

Comentário: O gabarito da questão é letra a), pois o comando DELETE remove uma ou mais linhas de uma tabela ou exibição no SQL Server.

A cláusula WHERE especifica as condições usadas para limitar o número de linhas que são excluídas. Se uma cláusula WHERE não for fornecida, DELETE removerá todas as linhas da tabela, contrariando o que diz a letra b).

Vamos às demais assertivas:

O comando TRUNCATE TABLE (Transact-SQL) remove todas as linhas de uma tabela ou partições especificadas de uma tabela sem registrar as exclusões de linha individual. TRUNCATE TABLE é semelhante à instrução DELETE sem nenhuma cláusula WHERE; entretanto, TRUNCATE TABLE é mais rápida e utiliza menos recursos de sistema e log de transações.

A instrução DELETE não apresenta a cláusula UP (n). Diferentemente, o comando apresenta o argumento TOP (expression) [PERCENT], que especifica o número ou a porcentagem de linhas aleatórias que serão excluídas. expression pode ser um número ou uma porcentagem das linhas. As linhas referenciadas na expressão TOP usada com INSERT, UPDATE ou DELETE não são organizadas em qualquer ordem.

A instrução TRUNCATE TABLE não apresenta a cláusula UP (n).

Gabarito: A

21.FCC - Analista Judiciário (TST)/Apoio Especializado/Análise de Sistemas/2017

Um Analista de Sistemas deseja fazer um backup completo de um banco de dados SQL Server chamado vendas para um disco cujo caminho é definido por 'Z:\servidor_backup\vendas.bak', formatando a mídia e comprimindo o banco de dados, utilizando Transact-SQL. Para isso, terá que utilizar o comando

- a) `RMAN DATABASE vendas to 'Z:\servidor_backup\vendas.bak' WITH FORMAT, COMPRESSION;`
- b) `BACKUP DATABASE vendas TO DISK = 'Z:\servidor_backup\vendas.bak' WITH FORMAT, COMPRESSION;`
- c) `BACKUP FROM vendas TO Z:\servidor_backup\vendas.bak SET FORMAT, COMPRESSION;`
- d) `BACKUP vendas TO Z:\servidor_backup\vendas.bak -F -C`



Comentário: A instrução BACKUP (Transact-SQL) faz o backup de um Banco de Dados SQL. O SQL Server faz backup de um banco de dados completo do SQL Server para criar um backup de banco de dados ou um ou mais arquivos ou grupos de arquivos do banco de dados para criar um backup de arquivo (BACKUP DATABASE). Além disso, no modelo de recuperação completa ou no modelo de recuperação bulk-logged, faz o backup do log de transações do banco de dados para criar um backup de log (BACKUP LOG). Veja a sintaxe para fazer um backup de um Database completo:

```
--Backing Up a Whole Database
BACKUP DATABASE { database_name | @database_name_var }
  TO <backup_device> [ ,...n ]
  [ <MIRROR TO clause> ] [ next-mirror-to ]
  [ WITH { DIFFERENTIAL
        | <general_WITH_options> [ ,...n ] } ]
[;]
```

As Opções WITH especificam opções a serem usadas com uma operação de backup. No caso da questão, temos:

FORMAT: Especifica que um novo conjunto de mídias deve ser criado. FORMAT faz com que a operação de backup grave um novo cabeçalho de mídia em todos os volumes de mídia usados para a operação de backup.

COMPRESSION: Habilita explicitamente a compactação de backup. Assim, temos a resposta na letra b):

```
BACKUP DATABASE vendas TO DISK = 'Z:\servidor_backup\vendas.bak' WITH FORMAT,
COMPRESSION;
```

Gabarito: B

22. CESPE - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

A ferramenta SQL Server Configuration Manager permite realizar configurações de modo que uma instância do SQL Server se inicie automaticamente quando o servidor for ligado.

Certo

Errado

Comentário: O gabarito é correto, pois o SQL Server Configuration Manager é uma ferramenta para gerenciar os serviços associados ao SQL Server, configurar os protocolos de rede usados pelo SQL Server para gerenciar a configuração de conectividade de rede de computadores cliente do SQL Server.

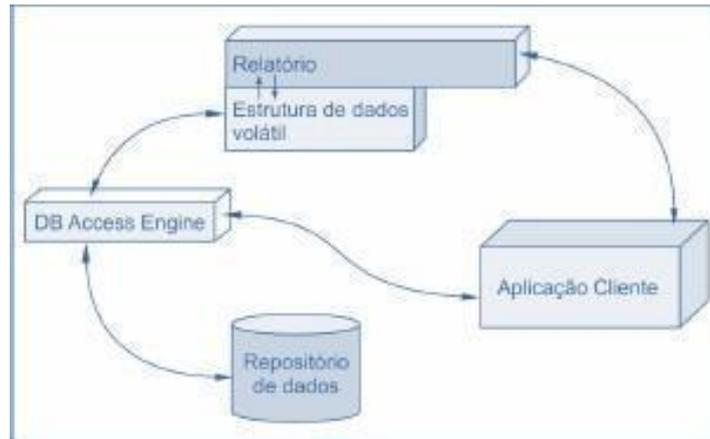
Durante a instalação, o SQL Server normalmente é configurado que a instância do SQL Server inicie automaticamente. Se isto não foi feito, você poderá alterar essa definição a qualquer momento.

Gabarito: C



A figura abaixo apresenta o diagrama da relação entre os principais elementos de uma solução baseada em SQL Server Reporting Services.





É correto afirmar sobre os elementos do diagrama:

- Os Relatórios podem ser de diversos tipos como: de análise, detalhado, de clickthrough, sub-relatório, histórico, armazenado em cache, instantâneo e relatório modelo.
- As Aplicações Cliente são capazes de implementar, gerenciar e visualizar os Relatórios, sendo os principais o Report XML Manager, Visual Studio, aplicações .NET por meio do ReportRDLX, além do aplicativo Office-SSRS Builder.
- Os Relatórios são baseados na linguagem RDLX (Report Definition XML Language), que é uma representação de XHTML criada pela Microsoft.
- DB Access Engine são os diversos mecanismos possíveis para acessar os dados. Corresponde às fontes de dados de acesso ao SQL Server e ao Microsoft Access, únicos SGBDs acessíveis.
- Repositório de Dados são as fontes de dados das quais se extraem as informações para os Relatórios criados e acessados através da linguagem SSRS.

Comentário: O SQL Server Reporting Services é uma solução local que os clientes implantam para criar, publicar, gerenciar relatórios e entregá-los aos usuários corretos de diferentes maneiras: exibindo-os em um navegador da Web, em seus dispositivos móveis ou como um email em suas caixas de entrada.

O SQL Server Reporting Services oferece um pacote atualizado de produtos:

Relatórios paginados "Tradicional" atualizados, de forma que você possa criar relatórios de aparência moderna, com ferramentas atualizadas e novos recursos para criá-los.

Novos relatórios móveis com um layout dinâmico que se adapta a diferentes dispositivos e as diferentes maneiras que você os segura.

Um portal da Web moderno que você pode exibir em qualquer navegador moderno. No novo portal, organize e exiba relatórios e KPIs móveis e paginados do Reporting Services. Também armazene pastas de trabalho do Excel no portal.

É possível criar relatórios paginados com tabela, matriz, gráfico e layouts de relatório de forma livre. Também é possível criar relatórios de tabelas para dados baseados em colunas, relatórios de matriz (como relatórios de tabela de referência cruzada ou de Tabela Dinâmica) para dados

O SQL Server permite usar várias fontes de dados para gerar relatórios, usando dados de qualquer tipo de fonte de dados que tenham um provedor de dados gerenciado por Microsoft .NET Framework, um provedor OLE DB ou uma fonte de dados ODBC. É possível ainda criar relatórios que usam dados relacionais e multidimensionais do SQL Server e do Analysis Services, Oracle, Hyperion e outros bancos de dados. Você pode usar uma extensão de processamento de dados XML para recuperar dados de qualquer fonte de dados XML. Também é possível usar funções com valor de tabela para projetar fontes de dados personalizadas.

Temos, então, o gabarito da questão na letra a).

Gabarito: A

24. CESPE - Analista Judiciário (TJDFT)/Apoio Especializado/Suporte em Tecnologia da Informação/2015

A respeito da configuração e da administração de sistemas gerenciadores de bancos de dados (SGBD) e de produtos a eles relacionados, julgue o item a seguir.

Apenas instalações autônomas de SQL Server permitem o uso de servidor de arquivos SMB como opção de armazenamento.

Certo

Errado

Comentário: A partir do SQL Server 2012 (11.x), os bancos de dados do sistema (Mestre, Modelo, MSDB e TempDB) e os bancos de dados de usuário do Mecanismo de Banco de Dados podem ser instalados com um servidor de arquivos SMB (protocolo SMB) como uma opção de armazenamento. Isso se aplica a instalações autônomas do SQL Server e a FCI (instalações de cluster de failover) do SQL Server. Assim, o gabarito é errado.

Gabarito: E

25. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

O comando do Transact SQL do Microsoft SQL Server 2008 para criar um sinônimo, atribuindo a denominação 'teste' à tabela 'primeiros_programas', do banco de dados 'primeiro_db' é

- a) SYNONYM teste EQUAL primeiro_db.primeiros_programas
- b) CREATE SYNONYM teste FOR primeiro_db.primeiros_programas
- c) SYNONYM teste ← primeiro_db.primeiros_programas
- d) DESCRIBE SYNONYM teste FOR EACH primeiro_db.primeiros_programas
- e) MAKE SYNONYM teste OF primeiro_db.primeiros_programas

Comentário: A instrução CREATE SYNONYM (Transact-SQL) cria um novo sinônimo. Veja a sintaxe:



```
CREATE SYNONYM [ schema_name_1. ] synonym_name FOR <object>

<object> :: =
{
  [ server_name.[ database_name ] . [ schema_name_2 ]. object_name
  | database_name . [ schema_name_2 ].| schema_name_2. ] object_name
}
```

synonym_name: é o nome do novo sinônimo.

object_name: é o nome do objeto base que o sinônimo referencia.

Temos, então, a resposta na letra b).

Gabarito: B

26. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

O comando do Transact SQL do Microsoft SQL Server 2008 para atualizar o valor de 20 registros, da coluna 'taxa', da tabela 'Blue', dividindo por 2 o valor dessa coluna 'taxa' é

- a) UPDATE MOST(20) Blue MAKE taxa = taxa/2
- b) UPDATE UPPER(20) Blue HAVING taxa = taxa/2.
- c) UPDATE SUP(20) Blue WITH taxa = taxa/2
- d) UPDATE TOP(20) Blue SET taxa = taxa/2
- e) UPDATE FIRST(20) Blue PRINTING taxa = taxa/2

Comentário: A instrução UPDATE (Transact-SQL) altera dados existentes em uma tabela ou exibição no SQL Server 2017. Veja a sintaxe:

```
[ WITH <common_table_expression> [...n] ]
UPDATE
  [ TOP ( expression ) [ PERCENT ] ]
  { { table_alias | <object> | rowset_function_limited
    [ WITH ( <Table_Hint_Limited> [ ...n ] ) ]
  }
  | @table_variable
}
SET
  { column_name = { expression | DEFAULT | NULL }
    | { udt_column_name. { property_name = expression
                        | field_name = expression }
      | method_name ( argument [ ,...n ] )
    }
  }
  | column_name { .WRITE ( expression , @Offset , @Length ) }
  | @variable = expression
  | @variable = column = expression
  | column_name { += | -= | *= | /= | %= | &= | ^= | |= } expression
  | @variable { += | -= | *= | /= | %= | &= | ^= | |= } expression
```



TOP (expression) [PERCENT]: Especifica o número ou o percentual de linhas atualizadas. expression pode ser um número ou uma porcentagem das linhas.

SET: Especifica a lista de colunas ou nomes de variáveis a serem atualizados.

column_name: é uma coluna que contém os dados a serem alterados. column_name precisa existir em table_or view_name. Colunas de identidade não podem ser atualizadas.

Expressão: é uma variável, valor literal, expressão ou uma instrução de subseleção (incluída com parênteses) que retorna um único valor. O valor retornado pela expressão substituirá o valor existente em column_name ou em *@variable*.

Assim, temos o gabarito da questão na letra d).

Gabarito: D

27.FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

No sistema gerenciador de banco de dados Microsoft SQL Server 2008, por meio da função sys.dm_index_physical_stats é possível verificar

- a) o número total de tabelas do banco de dados.
- b) a quantidade máxima de usuários simultâneos no banco de dados.
- c) o número total de usuários do banco de dados.
- d) a quantidade máxima permitida de tabelas no banco de dados.
- e) o nível de fragmentação de índices do banco de dados.

Comentário: A função sys.dm_db_fts_index_physical_stats (Transact-SQL) retorna uma linha para cada índice de texto completo ou semântico em cada tabela que tenha um índice de texto completo ou semântico associado.

| Nome da coluna | Tipo | Descrição |
|----------------------------|--------|--|
| object_id | INT | ID do objeto da tabela que contém o índice. |
| fulltext_index_page_count | bigint | Tamanho lógico da extração em número de páginas de índice. |
| keyphrase_index_page_count | bigint | Tamanho lógico da extração em número de páginas de índice. |



Assim, considerando o nome da coluna `object_id`, temos o ID do objeto da tabela que contém o índice. Por esse motivo, o gabarito da questão é a letra e).

Gabarito: E

28. CESPE - Analista Administrativo (ANTAQ)/TI - Analista de Infraestrutura/2014

Acerca do Microsoft SQL Server 2008, julgue o seguinte item.

O argumento `clustered` do comando `create index` cria um índice em que a ordem lógica dos valores da chave determina a ordem física das linhas correspondentes em uma tabela.

Certo

Errado

Comentário: A instrução `CREATE INDEX` (Transact-SQL) Cria um índice relacional em uma tabela ou exibição. Também chamado de um índice rowstore porque é um índice de árvore B clusterizado ou não clusterizado. Veja a sintaxe:

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name
ON <object> ( column [ ASC | DESC ] [ ,...n ] )
[ INCLUDE ( column_name [ ,...n ] ) ]
[ WHERE <filter_predicate> ]
[ WITH ( <relational_index_option> [ ,...n ] ) ]
[ ON { partition_scheme_name ( column_name )
      | filegroup_name
      | default
    } ]
[ FILESTREAM_ON { filestream_filegroup_name | partition_scheme_name | "NULL" } ]
```

O argumento `CLUSTERED` cria um índice no qual a **ordem lógica dos valores de chave determina a ordem física das linhas** correspondentes em uma tabela. O nível inferior, ou folha, do índice clusterizado contém as linhas de dados reais da tabela. Uma tabela ou exibição pode ter um índice clusterizado por vez.

Uma exibição com um índice clusterizado exclusivo é chamada de exibição indexada. Criar um índice clusterizado exclusivo em uma exibição materializa fisicamente a exibição. Um índice clusterizado exclusivo deve ser criado em uma exibição para que qualquer outro índice possa ser definido na mesma exibição.

Se `CLUSTERED` não for especificado, um índice não clusterizado será criado.

Dessa forma, o gabarito da questão é certo.

Gabarito: C

29. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2013

Um recurso presente no MS-SQL Server 2008 modela as entidades relacionadas ao SQL Server de uma organização em uma exibição unificada. O Gerenciador do Utilitário e os pontos de vista deste recurso no SQL Server Management Studio fornecem aos administradores uma exibição holística da integridade dos recursos do SQL Server por meio



- Instâncias do SQL Server.
- Aplicativos da camada de dados.
- Arquivos de banco de dados.
- Volumes de armazenamento.

Este recurso do SQL Server é chamado de

- a) SQL Entity Manager.
- b) UCP Manager.
- c) SQL Server Utility.
- d) SQL Query Administrator.
- e) Server Control Panel.

Comentário: O Utilitário do SQL Server modela as entidades relacionadas ao SQL Server de uma organização em uma exibição unificada. O Gerenciador do Utilitário e os pontos de vista do Utilitário do SQL Server no SSMS (SQL Server Management Studio) fornecem aos administradores uma visão holística da integridade dos recursos do SQL Server por meio de uma instância do SQL Server que funciona como um UCP (ponto de controle do utilitário).

A combinação de dados resumidos e detalhados apresentada no UCP para políticas de subutilização e de superutilização, e para uma variedade de parâmetros chave, permite que oportunidades de consolidação de recursos e de superutilização de recursos sejam identificadas facilmente. As políticas de integridade são configuráveis e podem ser ajustadas para alterar os limites inferior e superior da utilização de recursos. É possível alterar as políticas de monitoramento globais ou configurar políticas de monitoramento individuais para cada entidade gerenciada no Utilitário SQL Server.

Assim, temos o gabarito na letra c).

Gabarito: C

30. FCC - Agente de Defensoria Pública (DPE SP)/Analista de Sistemas/2013

Uma das funções disponíveis no sistema gerenciador de bancos de dados Microsoft SQL Server 2008 é @@CONNECTIONS. O resultado da aplicação dessa função é a obtenção

- a) do número de acessos autorizados, desde quando o SQL Server foi iniciado pela última vez.
- b) do número de tentativas de conexão bem ou mal sucedidas, desde quando o SQL Server foi iniciado pela última vez.
- c) do número de usuários simultâneos conectados ao SQL Server, no momento da execução dessa função.
- d) do tempo total de conexão de todos os usuários simultâneos do SQL Server, no momento da execução dessa função.
- e) da velocidade de conexão do usuário corrente ao Microsoft SQL Server.



@@MAX_CONNECTIONS é o número máximo permitido de conexões simultâneas com o servidor. @@CONNECTIONS é incrementado a cada tentativa de logon e, portanto, @@CONNECTIONS pode exceder @@MAX_CONNECTIONS.

Assim, temos o gabarito na letra b).

Gabarito: B

31. CESPE - Analista do Banco Central do Brasil/Área 5 - Infraestrutura e Logística/2013

A respeito do SQL Server 2012 Reporting Services e do Microsoft Office 2010, julgue o item subsequente.

Power View é um recurso do Suplemento SQL Server 2012 Reporting Services para Microsoft SharePoint Server 2010 Enterprise Edition. Esse recurso permite criar relatórios ad hoc intuitivos para usuários comerciais, além de criar e interagir com exibição de dados de modelos de dados com base em pastas de trabalho PowerPivot.

Certo

Errado

Comentário: Power View oferece relatórios ad hoc intuitivos para usuários comerciais, tais como analistas de dados, tomadores de decisões de negócios e operadores de informações. Eles podem criar e interagir facilmente com exibição de dados de modelos tabulares com base em pastas de trabalho Power Pivot publicadas em uma Galeria do Power Pivot ou modelos de tabela criados usando o SQL Server Data Tools (SSDT) e, em seguida, implantados em instâncias do SQL Server 2012 Analysis Services. Power View é um aplicativo do Silverlight baseado no navegador, iniciado no SharePoint Server 2010 ou posterior.

Ao criar projetos de modelo tabular no SQL Server Data Tools, você pode configurar determinadas propriedades de relatório exclusivas para relatórios do Power View.

Assim, o gabarito é certo.

Gabarito: C

32. CESPE - Analista Administrativo (TCE-ES)/Informática/2013

A ferramenta presente no SQL Server que, após analisar uma carga de trabalho, pode recomendar a adição, remoção ou modificação de estruturas de design físicas em bancos de dados é denominada

- SQL Server Studio.
- SQL Server Admin.
- SQL Server Configuration Manager.
- Orientador de Otimização do Mecanismo de Banco de Dados.
- SQL Server Profiler.

Comentário: O DTA (Orientador de Otimização do Mecanismo de Banco de Dados) do Microsoft



precisar de conhecimentos avançados sobre a estrutura do banco de dados ou dos recursos internos do SQL Server. Com o DTA, é possível executar as tarefas a seguir.

- Solucionar problemas de desempenho de uma consulta de problema específica
- Ajustar um conjunto grande de consultas por um ou mais bancos de dados
- Executar uma análise E-Se exploratória de possíveis alterações de design físico
- Gerenciar o espaço de armazenamento

Portanto, o gabarito é a letra d).

Gabarito: D

33. Ano: 2017 Banca: FGV Órgão: Alerj Cargo: Analista de Tecnologia da Informação Q. 50

No SQL Server 2012, os seguintes comandos foram executados individualmente para criar as tabelas no banco de dados MeuBanco.

```
CREATE TABLE Tab1
    (Codigo int PRIMARY KEY NOT NULL,
     Descricao text NULL);

CREATE TABLE Tab2
    (Cod int PRIMARY KEY NOT NULL,
     Descricao text NULL,
     Fonte int NOT NULL,
     CONSTRAINT FK_Tab2_Fonte FOREIGN KEY
 (Fonte) REFERENCES Tab1 (Codigo));

CREATE TABLE Tab3
    (ID_Seq int PRIMARY KEY NOT NULL,
     Inscricao int UNIQUE NOT NULL,
     Responsavel int NULL,
     CONSTRAINT FK_Tab3_Responsavel FOREIGN KEY
 (Responsavel) REFERENCES Tab3 (Inscricao));
```

A figura abaixo representa o conteúdo das tabelas Tab1, Tab2 e Tab3 de MeuBanco.



Tab1

| Codigo | Descricao |
|--------|-----------|
| 10 | Alfa |
| 20 | Beta |
| 30 | Qui |

Tab2

| Cod | Descricao | Fonte |
|------|-----------|-------|
| 1000 | Zeta | 30 |
| 2000 | Psi | 10 |
| 3000 | Xi | 20 |
| 4000 | Omega | 10 |
| 5000 | Upsilon | 20 |
| 6000 | Tau | 30 |

Tab3

| ID_Seq | Inscricao | Responsavel |
|--------|-----------|-------------|
| 1 | 201601 | 201603 |
| 2 | 201602 | 201602 |
| 3 | 201603 | 201603 |
| 4 | 201604 | 201604 |
| 5 | 201605 | 201603 |
| 6 | 201606 | 201602 |

Em momento posterior, os comandos abaixo foram executados individualmente na seguinte ordem:

```
TRUNCATE TABLE Tab1;
```

```
DELETE TOP (2) FROM Tab2 WHERE Cod < 4000;
```

```
TRUNCATE TABLE Tab3;
```

Considere a execução de commit implícitos e desconsidere quaisquer comandos reconhecidos unicamente por aplicativos clientes para acesso aos bancos de dados do SQL Server 2012. Após a execução dos comandos, é correto afirmar que:

- as linhas de Tab1 foram removidas e foi registrada uma entrada no log de transações para cada linha excluída;
- as linhas, a estrutura da tabela, as colunas e as constraints de Tab1 foram removidas;
- Tab2 ficou vazia, mas sua estrutura não sofreu alterações;
- os dados de Tab3 foram removidos e suas respectivas páginas de dados foram desalocadas;
- Tab3 teve seus dados removidos e foi registrada uma entrada no log de transações para cada linha excluída.

Comentário: O comando TRUNCATE vai remover todas as linhas da tabela e suas respectivas páginas sem gravar essa modificação nos arquivos de log. A estrutura da tabela permanece disponível no SGBD, agora sem dados armazenados. Sendo assim podemos marcar nossa resposta na alternativa D.



Gabarito: D

34. BANCA: VUNESP ANO: 2012 ÓRGÃO: TJ-SP PROVA: ANALISTA JUDICIÁRIO -
COMUNICAÇÃO E PROCESSAMENTO DE DADOS

No Microsoft SQL Server 2008, as teclas de atalho para inserir ou remover um bookmark são:

A Ctrl + K

B Ctrl + B

C Alt + U

D Alt + K

E Alt + B

Comentário. O SQL Server possui um recurso de marcadores, indicadores ou bookmarks. Para utilizar esses marcadores você pode fazer uso das teclas de atalho. No caso específico de inserir ou remover um indicador você deve usar o Ctrl + K. Segue abaixo uma lista com as teclas de atalho para bookmarks do SQL Server. Uma lista de outros atalhos pode ser vista [aqui](#).

| Ação | SQL Server 2014 | SQL Server 2008 R2 |
|---|---|--------------------|
| Definir ou remover um indicador na linha atual | CTRL+K, CTRL+K | CTRL+K, CTRL+K |
| Próximo indicador | CTRL+K, CTRL+N | CTRL+K, CTRL+N |
| Se o indicador atual estiver em uma pasta, moverá para o próximo indicador na pasta. Os indicadores fora da pasta são ignorados. Se o indicador não estiver em uma pasta, moverá para o próximo indicador no mesmo nível. Se a janela Indicadores contiver uma pasta, os indicadores em pastas serão ignorados. | CTRL+SHIFT+K, CTRL+SHIFT+N CTRL+SHIFT+P | Sem equivalente |
| Indicador anterior | CTRL+K, CTRL+P | CTRL+K, CTRL+P |
| Limpar indicadores | CTRL+K, CTRL+L | CTRL+K, CTRL+L |

Gabarito: A

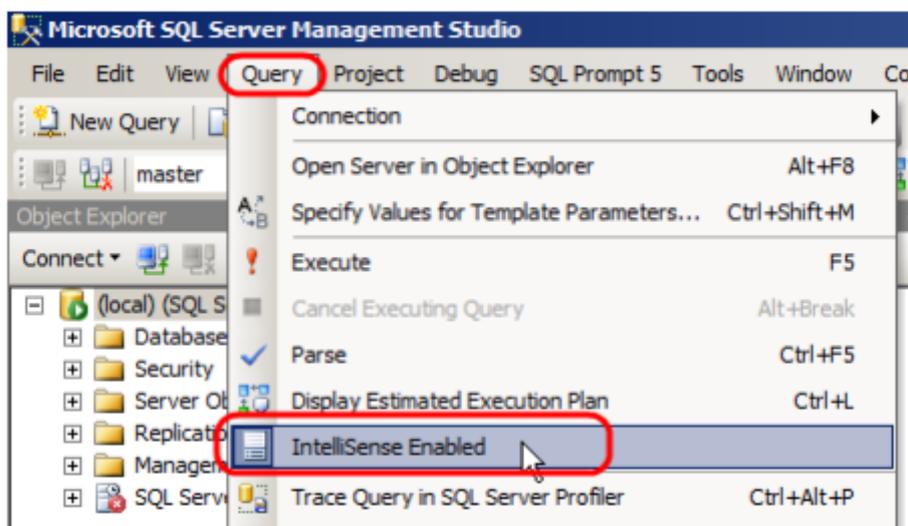


35.BANCA: VUNESP ANO: 2013 ÓRGÃO: IMESC PROVA:
ANALISTA DE TECNOLOGIA - INFORMÁTICA

O sistema gerenciador de bancos de dados Microsoft SQL Server 2008 conta com um recurso denominado IntelliSense que provê facilidades para a edição de comandos. O botão do SQL Server 2008 que permite habilitar ou desabilitar esse recurso é

- A 
- B 
- C 
- D 
- E 

Comentário. O editor do SQL Server Management Studio suporta opções do Microsoft IntelliSense que reduzem a digitação, fornecem acesso rápido a informações de sintaxe, ou torna mais fácil visualizar os delimitadores de expressões complexas. O Microsoft IntelliSense fornece uma variedade de opções que fazem as referências à linguagem acessíveis de forma mais fácil. Durante a codificação, você não precisa deixar o editor realizar pesquisas sobre os elementos de linguagem. Você pode manter seu contexto, encontrar a informação que você precisa, inserir elementos da linguagem diretamente em seu código, e ainda deixar o IntelliSense completar a sua digitação. Veja abaixo a figura que mostra o ícone para habilitar da ferramenta:



Gabarito: D

36. BANCA: VUNESP ANO: 2012 ÓRGÃO: TJ-SP PROVA: ANALISTA JUDICIÁRIO - COMUNICAÇÃO E PROCESSAMENTO DE DADOS

Um usuário do Microsoft SQL Server 2008 deseja atribuir um sinônimo de nome S1 para a tabela Produtos. O código do Transact SQL para executar essa função é

- A ATTRIB SYM S1 FOR Produtos;
- B ATTRIB SYNONIM Produtos (S1);
- C CREATE SYNONYM S1 FOR Produtos;
- D CREATE SYNM S1 (Produtos);
- E CREATE SYM S1 Produtos;

Comentário. Um sinônimo é um objeto de banco de dados que atende aos seguintes propósitos. Primeiramente, fornece um nome alternativo para outro objeto do banco de dados referido como o objeto base que pode existir em um servidor local ou remoto. Ele também fornece uma camada de abstração que protege um aplicativo cliente de alterações feitas no nome ou local do objeto base.

Por exemplo, considere a tabela Employee do Adventure Works localizado em um servidor denominado Server1. Para fazer referência a esta tabela em outro servidor, Server2, um aplicativo precisa usar o nome de quatro partes Server1.AdventureWorks.Person.Employee. Além disso, se o local da tabela for alterado, por exemplo, para outro servidor, o aplicativo cliente precisará ser modificado para refletir essa alteração.

Para resolver os dois problemas, é possível criar um sinônimo, EmpTable, no Server2 para a tabela Employee no Server1. Agora, o aplicativo cliente precisa usar apenas o nome de uma única parte, EmpTable, para fazer referência à tabela Employee. Além disso, se o local da tabela Employee for alterado, você precisará modificar o sinônimo, EmpTable, para apontar para o novo local da tabela Employee. Como não existe nenhuma instrução ALTER SYNONYM, você precisa primeiro descartar o sinônimo, EmpTable, e recriá-lo com o mesmo nome, mas apontá-lo para o novo local de Employee.

Veja abaixo um exemplo de criação e uso de um sinônimo.

```
USE tempdb;
GO
-- Create a synonym for the Product table in AdventureWorks2012.
CREATE SYNONYM MyProduct
FOR AdventureWorks2012.Production.Product;
GO

-- Query the Product table by using the synonym.
USE tempdb;
GO
SELECT ProductID, Name
FROM MyProduct
WHERE ProductID < 5;
GO
```

Você pode conhecer um pouco mais sobre sinônimos [aqui](#).



Gabarito: C

37. BANCA: VUNESP ANO: 2013 ÓRGÃO: CETESB PROVA: ANALISTA DE TECNOLOGIA DA INFORMAÇÃO - ADMINISTRADOR DE BANCO DE DADOS

O procedimento armazenado do Transact do SQL Microsoft SQL Server 2008 que exibe informações sobre as dependências entre objetos (tabelas, visões, ...) de um banco de dados é

- A sp_lock.
- B sp_fkeys.
- C sp_depends.
- D sp_enumdsn.
- E sp_grantlogin.

Comentário. O SQL Server prover uma lista de procedimentos armazenados que provêm informações sobre os objetos presentes no banco de dados. São conhecidos como System Stored Procedures. Vamos analisar cada uma das alternativas presentes na questão:

sp_lock – Retorna informações sobre locks. Essa função será removida em versões futuras do SQL Server. Evite utilizar. Para obter informações sobre bloqueios use a visão dinâmica sys.dm_tran_locks.

sp_fkeys – Exibe as informações lógicas sobre as chaves estrangeiras no ambiente atual. Essas procedures mostra os relacionamentos entre as chaves estrangeiras, inclusive as chaves estrangeiras desabilitadas.

sp_depends - Exibe informações sobre dependências de objetos de banco de dados, tais como os visões e procedimentos que dependem de uma tabela ou visão. As referências a objetos fora do banco de dados atual não são relatados.

sp_enumdsn - Retorna uma lista de todos os nomes de fontes e dados ODBC e OLE DB definidos para um servidor que está executando sob uma conta de usuário específico do Microsoft Windows. Esse procedimento armazenado é executado no Publisher em qualquer banco de dados.

sp_grantlogin – Cria um login no SQL Server. Essa funcionalidade será removida em versões futuras do SQL Sever. Em seu lugar você deve utilizar o comando CREATE LOGIN.

Gabarito: C

38. BANCA: VUNESP ANO: 2013 ÓRGÃO: CETESB PROVA: ANALISTA DE TECNOLOGIA DA INFORMAÇÃO - ADMINISTRADOR DE BANCO DE DADOS

Sobre os Sistemas Gerenciadores de Bancos de Dados (considerando o Microsoft SQL Server 2008), é correto afirmar que



A devem ser desligados por, pelo menos, 2 horas diariamente.

B devem tratar comandos emitidos pelo usuário, permitindo, por exemplo, a busca de dados.

C não podem ser utilizados em ambiente de rede.

D podem ser substituídos, sem perda de funcionalidade, por um programa de gerenciamento de tarefas.

E podem funcionar sem a presença de um sistema operacional no servidor.

Comentário. Essa é uma questão bem tranquila que não exige nenhum conhecimento aprofundado de SQL Server. Vejam que a nossa resposta, que é a alternativa B, traz uma frase que deve valer para qualquer banco de dados. Afinal de contas, executar consultas sobre a base é uma das principais tarefas de qualquer SGBD.

Gabarito: B

39. BANCA: VUNESP ANO: 2013 ÓRGÃO: COREN-SP PROVA: ANALISTA - ADMINISTRADOR DE BANCO DE DADOS

Considerando o tuning do sistema gerenciador de bancos de dados Microsoft SQL Server 2012, um dos parâmetros monitorados refere-se ao uso do disco rígido, que indica

A a porcentagem de tempo de ocupação do disco com operações de leitura e escrita.

B a razão entre o número de páginas lidas e escritas.

C a razão entre o número de registros inseridos e excluídos.

D o número de acessos por minuto ao disco rígido.

E o número médio de bytes lidos e escritos em um segundo.

Comentário. Sobre performance em banco de dado SQL Server existe uma infinidade de livros sobre o assunto. Não sendo possível tratar em detalhes aqui vamos procurar listar apenas as informações a respeito do monitoramento de disco feitas pelas ferramentas de tuning. Para o SQL Server para executar de maneira ótima, monitoramento e otimização do subsistema do SQL Server Disk é um dos aspectos importantes. Temos requisitos muito específicos de desempenho do disco.

Você pode usar os seguintes valores para monitorar o desempenho viando identificar corretamente os problemas de desempenho do disco.

PhysicalDisk Object: A média do Disk Queue Length representa o número médio de leitura física e pedidos de escrita que foram enfileiradas no disco durante o período de amostragem. Se o seu sistema de I/O está sobrecarregado, mais operações de leitura/gravação esperarão. Se o comprimento da fila de disco frequentemente excede um valor de 2, durante o pico de uso do SQL Server, você pode ter um gargalo de I/O.

AVG. Disk Sec/Read é o tempo médio, em segundos, de uma leitura de dados do disco. AVG. Disk Sec/Write é o tempo médio, em segundos, de uma gravação de dados no disco.

Disco físico: % Disk Time é a porcentagem de tempo decorrido que a unidade de disco



é inferior a 85 por cento da capacidade do disco.

AVG. Gravações de disco/s é a taxa de operações de gravação no disco. Certifique-se que este número é inferior a 85 por cento da capacidade do disco. O tempo de acesso ao disco aumenta exponencialmente para valores acima de 85 por cento da capacidade.

Observem que, pelo exposto acima, a única alternativa que fazer sentido é a presente na letra A.

Gabarito: A

40. BANCA: VUNESP ANO: 2014 ÓRGÃO: PRODEST-ES PROVA: ANALISTA DE TECNOLOGIA DA INFORMAÇÃO - DESENVOLVIMENTO DE SISTEMAS

No Transact SQL do sistema gerenciador de bancos de dados MS SQL Server 2008 R2, o comando geral para a criação de um procedimento armazenado é:

A CREATE PROCEDURE <nome do procedimento>

...

AS

<comando

SQL> GO

B FORM PROCEDURE <nome do procedimento>

...

OF

<comando

SQL> GO

C CREATE PROCEDURE <nome do procedimento>

...

LIKE

<comando SQL>

GO

D GENERATE PROCEDURE <nome do procedimento>

...

HOW

<comando SQL>

GO

E MAKE PROCEDURE <nome do procedimento>

...

OF TYPE

<comando SQL>

GO

Comentário. Além dos procedimentos armazenados de sistemas que falamos anteriormente, o usuário tem condições de criar suas próprias procedures. A questão pergunta qual a sintaxe básica



```
--SQL Server Stored Procedure Syntax
CREATE { PROC | PROCEDURE } [schema_name.] procedure_name [ ; number ]
    [ { @parameter [ type_schema_name. ] data_type }
      [ VARYING ] [ = default ] [ OUT | OUTPUT | [READONLY]
        ] [ ,...n ]
    [ WITH <procedure_option> [ ,...n ] ]
    [ FOR REPLICATION ]
AS { [ BEGIN ] sql_statement [;] [ ...n ] [ END ] }
[;]

<procedure_option> ::=
    [ ENCRYPTION ]
    [ RECOMPILE ]
    [ EXECUTE AS Clause ]
```

Podemos observar que a sintaxe converge com a resposta da questão na alternativa A.

Gabarito: A

41. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE TÉCNICO - DESENVOLVEDOR

O comando do sistema gerenciador de bancos de dados Microsoft SQL Server 2008 que possibilita remover o acesso de um usuário é:

- A DELETE MEN <nome>
- B DELETE MEMBER <nome>
- C DROP LOGIN <nome>
- D DROP ACTION <nome>
- E EXCEPT ACTION
<nome>

Comentário. Essa questão tem uma pegadinha. O examinador quer levar você a pensar nos comandos GRANT, REVOKE e DENY usados para garantir e retirar privilégios de acesso sobre objetos. Contudo, nenhuma das respostas possui essas alternativas.

Temos, portanto, que voltar nosso foco para outra opção que seria remover o acesso de um usuário por meio da exclusão do seu login. Isso é feito por meio do comando DROP LOGIN <nome>.

É importante salientar que um LOGIN não pode ser excluído enquanto o usuário estiver logado ao banco. Um LOGIN que pertença a algum job de segurança, server-level object, ou SQL Server Agent não pode ser apagado.

Para executar o comando precisamos da permissão de ALTER ANY LOGIN no servidor.

Gabarito: C

42. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE TÉCNICO - DESENVOLVEDOR



- A criar um trigger.
- B definir um ponto de salvamento de transações.
- C encerrar todas as conexões com o banco de dados.
- D fazer o backup de um banco de dados.
- E fazer o backup de uma tabela.

Comentário. O SQL Server prover os seguintes comandos para ajudar no gerenciamento de transações: BEGIN DISTRIBUTED TRANSACTION, ROLLBACK TRANSACTION, BEGIN TRANSACTION, ROLLBACK WORK, COMMIT TRANSACTION, SAVE TRANSACTION e COMMIT

WORK. Os dois comandos de BEGIN marcam explicitamente o início de uma transação, a diferença é que quando utilizamos a cláusula DISTRIBUTED devemos tratar de transações de distribuídas, quando não utilizada trata de uma transação local. O BEGIN TRANSACTION incrementa o @@TRANCOUNT em um.

O ROLLBACK TRANSACTION retorna uma transação, implícita ou explícita, para o início da transação ou para algum SAVEPOINT dentro da transação. O ROLLBACK WORK retorna uma transação especificada pelo usuário para o início da mesma.

O COMMIT TRANSACTION marca o fim da execução bem-sucedida de uma transação. Esse comando vai liberar os recursos bloqueados e decrementar a variável @@TRANCOUNT. O COMMIT WORK marca o fim de uma transação, funciona exatamente igual ao COMMIT TRANSACTION.

Por fim, o SAVE TRANSACTION que basicamente cria um savepoint dentro da transação. Esse ponto permite que a transação volte até ele em caso de falha. Observem, então, que ele tem por objetivo definir um ponto de salvamento de transações. Essa é a nossa resposta, presente na alternativa B.

Gabarito: B



43. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

Uma forma de inserir comentários em um comando SQL no sistema gerenciador de banco de dados Microsoft SQL Server 2012 é

- A @ texto do comentário
- B § texto do comentário
- C -- texto do comentário
- D % texto do comentário %
- E # texto do comentário #

Comentário. Existem basicamente duas formas de inserir comentários dentro de comandos do SQL Server. A primeira é utilizada quando você pretende comentar apenas uma linha, usa-se - para determinar que aquela linha é um comentário e, portanto, não será avaliada pelo servidor. A outra opção é a utilização do /* e */. Todo o texto inserido entre essas tags são considerados comentários, mesmo que ocupem mais de uma linha do arquivo. Vejam os exemplos abaixo para cada um dos tipos de comentários:

```
-- Choose the AdventureWorks2012 database.  
USE AdventureWorks2012;  
GO  
-- Choose all columns and all rows from the Address table.  
SELECT *  
FROM Person.Address  
ORDER BY PostalCode ASC; -- We do not have to specify ASC because  
-- that is the default.  
GO
```

```
USE AdventureWorks2012;  
GO  
/*  
This section of the code joins the Person table with the Address table,  
by using the Employee and BusinessEntityAddress tables in the middle to  
get a list of all the employees in the AdventureWorks2012 database  
and their contact information.  
*/  
SELECT p.FirstName, p.LastName, a.AddressLine1, a.AddressLine2, a.City, a.PostalCode  
FROM Person.Person AS p  
JOIN HumanResources.Employee AS e ON p.BusinessEntityID = e.BusinessEntityID  
JOIN Person.BusinessEntityAddress AS ea ON e.BusinessEntityID = ea.BusinessEntityID  
JOIN Person.Address AS a ON ea.AddressID = a.AddressID;  
GO
```

Gabarito: C



44. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

No sistema gerenciador de banco de dados Microsoft SQL Server 2008, o procedimento armazenado que exibe informações sobre os usuários atuais do banco de dados é

- A sp_who
- B sp_lock
- C sp_pkeys
- D sp_catalogs
- E sp_grantlogin.

Comentário. Vejamos o que cada um dos procedimentos armazenados descritos nas alternativas traz como informação no seu resultado.

O **sp_who** prover informações sobre os usuários correntes, as sessões e processos em execução na instância do motor do SQL Server.

O **sp_lock** reporta as informações sobre os bloqueios do banco de dados.

O **sp_pkeys** retorna a informação a respeito da chave primária de uma tabela específica. O **sp_catalogs** não existe dentro do rol de procedimento armazenados no SQL Server.

Por fim temos o **sp_grantlogin**, esse comando (embora deprecated) permite a criação de um login no SQL Server. Vejam que após a análise de cada uma das alternativas nossa resposta encontra-se na alternativa A.

Gabarito: A

45. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

No sistema gerenciador de banco de dados Microsoft SQL Server 2008, a seleção de @@TRANCOUNT tem como resultado o número

- A de tabelas com valores nulos.
- B de transações ativas na conexão atual.
- C de triggers disparados durante a conexão atual.
- D de usuários conectados ao servidor.
- E médio de registros por tabela.

Comentário. A seleção sobre a função @@TRANCOUNT retorna o número de comandos BEGIN TRANSACTION ativos na conexão atual no momento da consulta. Vejam, portanto, que confirmamos a resposta na alternativa B.

Gabarito: B



46. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

O comando do sistema gerenciador de banco de dados Microsoft SQL Server 2000 para parar imediatamente o gerenciador é

A KILL

B STOP

C LOCK

D REVOKE

E SHUTDOWN

Comentário. Se nós quisermos desligar o SQL Server, sem realizarmos pontos de verificação nos bancos de dados, e sem tentar terminar os processos de usuário, usamos o seguinte comando:

SHUTDOWN

Ao executar o comando acima, o SQL Server, irá parar, seguindo as seguintes etapas:

1. Desabilitar logons (exceto para os membros sysadmin e serveradmin).
2. Esperar a conclusão de procedimentos armazenados ou instruções TSQL em execução. Para exibir uma lista de todos os processos ativos e bloqueios, execute `sp_who` e `sp_lock`, respectivamente.
3. Inserir um ponto de verificação em cada banco de dado da instância.

Para encerrar o SQL server sem os pontos de verificação em cada banco de dados, execute com o argumento: "WITH NOWAIT".

SHUTDOWN WITH NOWAIT

As permissões de desligamento são atribuídas a membros de "sysadmin" e "ServerAdmin". Pelo exposto acima temos a nossa resposta na alternativa E.

Gabarito: E

47. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

O comando do sistema gerenciador de banco de dados Microsoft SQL Server 2000 que tem como retorno o número de leituras de disco, desde a última ativação do gerenciador é

A SELECT @@TOTAL_READ

B SELECT @@PROCID

C SELECT @@DBTS



D SELECT @@IDLE

E SELECT @@SPID

Comentário. Vamos analisar cada uma das alternativas. O @@TOTAL_READ retorna o número de leituras de disco, não leituras de cache, feitas pelo SQL Server desde a última vez que o servidor foi inicializado. Vejam que está é nossa resposta. Esta função está classificada dentro das funções estatísticas de sistema do SQL Server. Essas funções são não determinísticas, ou seja, não há garantia que retornem o mesmo valor de saída para duas entradas idênticas.

As outras funções presentes neste grupo são: O @@CONNECTIONS que retorna o número de tentativas de conexões, bem sucedidas ou não, desde o SQL Server foi iniciado pela última vez, @@PACK_RECEIVED que retorna o número de pacotes de entrada lido a partir da rede pelo SQL Server desde que ele foi iniciado, @@CPU_BUSY que retorna o tempo que o SQL Server passou trabalhando desde que foi iniciado pela última vez, @@PACK_SENT que retorna o número de pacotes de saída gravados na rede pelo SQL Server desde que ele foi iniciado pela última vez, fn_virtualfilestats que retorna estatísticas de entrada e saída (I/O) para arquivos de banco de dados, incluindo arquivos de log. Em SQL Server, essa informação também está disponível a partir da visão de gerenciamento dinâmico sys.dm_io_virtual_file_stats.

Temos ainda @@IDLE retorna o tempo ocioso desde a última inicialização. O @@TOTAL_ERRORS retorna o número total de erros. O @@IO_BUSY mostra o tempo que o SQL Server gasta fazendo operações de entrada e saída. O @@PACKET_ERRORS apresenta a quantidade de erros de pacotes de rede. O @@TOTAL_WRITE retorna o número de escritas em disco desde a última atualização. Observem que já tratamos da alternativa D.

Vamos continuar agora analisando a letra B. O @@PROID retorna o ID (identificador) do objeto do módulo Transact-SQL atual. Um módulo Transact-SQL pode ser um procedimento armazenado, uma função definida pelo usuário ou um gatilho. @@PROCID não pode ser especificado em módulos CLR ou no provedor de acesso de dados em processo.

A próxima alternativa é a C. @@DBTS retorna o valor do tipo de dados timestamp atual para o banco de dados atual. Este carimbo de data e hora é garantido como exclusivo no banco de dados.

Por fim o @@SPID que retorna o ID de sessão do processo de usuário atual.

Gabarito: A



LISTA DE QUESTÕES - CESPE (CEBRASPE)

1. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A unidade mais comum de medida de área é o hectare.

2. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A função ST_Area permite calcular a área de polígonos.

3. CEBRASPE (CESPE) - Especialista Técnico (BNB)/Analista de Sistema/2018

Acerca de bancos de dados, julgue o item que se segue.

O código a seguir, criado no SQL Server 2017, apresenta uma visão materializada, especificamente devido ao argumento SCHEMABINDING.

```
CREATE VIEW
```

```
VwTeste WITH
```

```
SCHEMABINDING
```

```
AS
```

```
SELECT campo1 FROM tabela WHERE campo1 > 17;
```

4. CEBRASPE (CESPE) - Professor de Educação Básica (SEDF)/Informática/2017

Julgue o item a seguir, a respeito de banco de dados, organização de arquivos, métodos de acesso e banco de dados textuais.

Ao executar consultas aninhadas, os bancos de dados SQL Server e DB2 utilizam avaliação correlacionada para eliminar as correlações, sem considerarem como opção o nivelamento das consultas aninhadas ou a utilização de técnicas de reescrita.

5. CEBRASPE (CESPE) - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

A ferramenta SQL Server Configuration Manager permite realizar configurações de modo que uma instância do SQL Server se inicie automaticamente quando o servidor for ligado.

6. CEBRASPE (CESPE) - Auditor de Controle Externo (TCE-PA)/Informática/Analista



de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

Caso a senha de uma conta do SQL Server 2008 R2 seja alterada, a nova senha entrará em vigor imediatamente, sem a necessidade de reinicialização do SQL Server.

7. Ano: 2015 Banca: CESPE Órgão: STJ Prova: Analista Judiciário - Infraestrutura

A respeito da configuração e administração de banco de dados, julgue os próximos itens.

[1] Diferentemente das versões anteriores, o SQL Server 2014 não pode ser instalado em computadores com sistema de arquivos FAT32, mas apenas em computadores com sistema de arquivos NTFS.

8. CEBRASPE (CESPE) - Inspetor de Controle Externo (TCE-RN)/Tecnologia da Informação/2015

No que se refere a tecnologia e arquitetura de banco de dados, julgue o próximo item.

No MSSQL Server 2014, o recurso AlwaysOn é uma solução de alta disponibilidade e de recuperação de desastres que fornece uma alternativa, em nível corporativo, para o espelhamento de bancos de dados, a partir do gerenciamento de réplicas de bancos de dados.

9. Ano: 2015 Banca: CESPE Órgão: MEC Prova: TÉCNICO DE NÍVEL SUPERIOR - ANALISTA DE SISTEMAS

Acerca dos sistemas gerenciadores de banco de dados (SGBD) PostgreSQL, Microsoft SQL Server e Oracle, julgue os itens a seguir.

Uma das principais novidades do Microsoft SQL Server 2014 é o recurso OLTP na memória (In- memory OLTP), o qual permite melhorar significativamente o desempenho de sistemas com processamento de transações on-line e data warehousing. A única maneira de remover um grupo de arquivos com otimização de memória é descartar o banco de dados.

10. CEBRASPE (CESPE) - Analista Judiciário (TJDFT)/Apoio Especializado/Suporte em Tecnologia da Informação/2015

A respeito da configuração e da administração de sistemas gerenciadores de bancos de dados (SGBD) e de produtos a eles relacionados, julgue o item a seguir.

Se, na modificação de determinada instância existente de SQL Server, for realizada a instalação de componentes de replicação, será necessário reiniciar o agente de SQL Server.

11. Ano: 2015 Banca: CESPE Órgão: MEC Prova: TÉCNICO DE NÍVEL SUPERIOR - ANALISTA DE SISTEMAS

Julgue os itens subsequentes, relativos ao Microsoft SQL Server.

[1] SQL Server fornece um conjunto de tipos de dados primitivos tipos de cadeia de strings de



tamanho fixo e variável até 2^{90} .

12. Ano: 2015 Banca: CESPE Órgão: MEC Prova: TÉCNICO DE NÍVEL SUPERIOR - ANALISTA DE SISTEMAS

Julgue os itens subsequentes, relativos ao Microsoft SQL Server.

[1] VIEW é uma tabela virtual cujo conteúdo está definido por uma instrução SELECT.

13. ANO: 2014 BANCA: CESPE ÓRGÃO: ANATEL PROVA: ANALISTA ADMINISTRATIVO - SUPORTE E INFRAESTRUTURA DE TI

A respeito de banco de dados, julgue os itens que se seguem.

[1] É válida para o PostgreSQL 9.3, mas não para o SQL Server 2012, a criação da SEQUENCE seqa por meio do seguinte comando:

```
CREATE SEQUENCE seqa START WITH 1;
```

14. ANO: 2014 BANCA: CESPE ÓRGÃO: ANTAQ PROVA: ANALISTA ADMINISTRATIVO - ANALISTA DE INFRAESTRUTURA

Acerca do Microsoft SQL Server 2008, julgue os seguintes itens.

[1] Para desabilitar uma trigger DDL (data definition language) definida com escopo de servidor (on all server), é necessária a permissão control server no servidor.

15. ANO: 2014 BANCA: CESPE ÓRGÃO: ANTAQ PROVA: ANALISTA ADMINISTRATIVO - ANALISTA DE INFRAESTRUTURA

Acerca do Microsoft SQL Server 2008, julgue os seguintes itens.

[1] O argumento clustered do comando create index cria um índice em que a ordem lógica dos valores da chave determina a ordem física das linhas correspondentes em uma tabela.

16. ANO: 2014 BANCA: CESPE ÓRGÃO: TC-DF PROVA: ANALISTA DE ADMINISTRAÇÃO PÚBLICA - SISTEMAS DE TECNOLOGIA DA INFORMAÇÃO

Julgue os seguintes itens, acerca de sistemas de gerenciamento de bancos de dados (SGBD) e de cópias de segurança de dados.

[1] No SQL Server 2012, a criação de índices em tabelas temporárias locais pode ser feito off-line, desde que a tabela não possua tipos de dados LOB (Large Object).

17. CEBRASPE (CESPE) - Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

No Sql Server, um becape

a) de arquivo agrega um ou mais arquivos ou grupos de arquivos de banco de dados.

b) parcial agrega dados que ainda não foram afetados por COMMIT, tanto de dados



quanto de transações.

- c) completo de banco de dados agrega todos os dados de todos os bancos de dados no momento em que o becape é concluído, com exceção dos logs de transação.
- d) diferencial agrega apenas logs de transações, incluindo somente transações feitas desde seu último becape de log até a transação mais recente.
- e) de logs de transações inclui todos os registros de log de forma cumulativa, independentemente de ter havido becape de log anterior ou becape completo.

18. CEBRASPE (CESPE) - Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

Acerca das soluções presentes no Sql Server 2008 R2, assinale a opção correta.

- a) Reporting services fornece uma plataforma de comunicação embasada em mensagens assíncronas que permite a interoperabilidade entre sistemas.
- b) Integration services permite aos usuários integrar e gerenciar estruturas multidimensionais que contenham dados agregados de outras fontes de dados, como bancos de dados relacionais.
- c) SharePoint services é uma solução utilizada para construir soluções de integração de dados, incluindo a extração, transformação e carregamento de dados.
- d) Master data services visa integrar sistemas analíticos e operacionais distintos de modo que seja criada e gerenciada uma fonte de informações central e auditável.
- e) Service broker é uma ferramenta voltada para a criação de cluster de alta disponibilidade, permitindo que os dados sejam acessados e recuperados de forma distribuída, e, em caso de indisponibilidade em um dos nós da solução cluster, há recuperação automática dos dados.

19. CEBRASPE (CESPE) - Analista Judiciário (TRE MS)/Apoio Especializado/Análise de Sistemas/2013

Acerca de tuning de banco de dados, bem como de técnicas de análise de desempenho e otimização de consultas em SQL Server 2008 R2, assinale a opção correta.

- a) Se, no plano de execução de determinada consulta, o DBA visualizou um ícone na execução da consulta representando a operação bitmap, então a consulta encontrou, em determinado momento, um campo do tipo BLOB (binary large object) ou um índice do tipo fulltext.
- b) A opção fillfactor no índice com valor 0 indica que não haverá folheamento de índices, o que se traduz em velocidade, ao passo que, no índice com valor 100, essa opção indica que não haverá splits, porém maior necessidade de espaço para armazenamento de dados.
- c) Se o Sql Server estiver instalado em uma máquina com mais de um processador, ele pode decidir a quantidade de processadores a serem utilizados no processamento da consulta, cujo valor é limitado ao número máximo de processadores conforme configurado na opção max degree of parallelism. A opção MAXDOP na criação de um índice pode mudar essa configuração durante a execução de tal índice.



d) Uma tabela que possua uma chave primária composta exige que seus dados sejam fisicamente classificados e armazenados com base em seus valores de chave e, desse modo, por padrão, o Sql Server cria um índice clusterizado (CLUSTERED) na tabela para cada atributo pertencente à chave primária.

e) Se, no plano de execução de determinada consulta, o DBA visualizou um ícone na execução da consulta representando a operação the nested loops, então a consulta utilizou, em determinado momento, um indicador de fila ou chave de clustering para pesquisar a linha correspondente na tabela ou índice clusterizado.

20. ANO: 2013 BANCA: CESPE ÓRGÃO: ANTT PROVA: ANALISTA ADMINISTRATIVO - INFRAESTRUTURA DE TI

A respeito de SQL Server, julgue os itens subsecutivos.

[1] Um dos recursos da LINQ (language integrated query) disponíveis no SQL Server 2008 é a realização de consultas, pelo desenvolvedor, diretamente em base de dados via framework.

21. CESPE - Analista Administrativo (TCE-ES)/Informática/2013

A ferramenta presente no SQL Server que, após analisar uma carga de trabalho, pode recomendar a adição, remoção ou modificação de estruturas de design físicas em bancos de dados é denominada

- a) SQL Server Studio.
- b) SQL Server Admin.
- c) SQL Server Configuration Manager.
- d) Orientador de Otimização do Mecanismo de Banco de Dados.
- e) SQL Server Profiler.

22. CEBRASPE (CESPE) - Analista do Ministério Público da União/Tecnologia da Informação e Comunicação/Suporte e Infraestrutura/2013

Julgue o próximo item, acerca dos sistemas ORACLE, MySQL e SQL Server.

Na instalação do SQL Server 2012 em um sistema de arquivos NTFS, são definidas automaticamente as ACL (access control lists) em chaves de registro e em arquivos, devendo as ACL ser revisadas imediatamente após a conclusão da instalação.

23. ANO: 2013 BANCA: CESPE ÓRGÃO: STF PROVA: ANALISTA JUDICIÁRIO - SUPORTE EM TECNOLOGIA DA INFORMAÇÃO

A respeito de configuração e administração de bancos de dados, julgue os itens a seguir.

[1] No processo de instalação do SQL Server 2012, deve-se instalar apenas uma cópia das ferramentas de gerenciamento, independentemente da quantidade de instâncias do SQL Server instaladas na máquina.

24. ANO: 2013 BANCA: CESPE ÓRGÃO: STF PROVA: ANALISTA JUDICIÁRIO - SUPORTE EM TECNOLOGIA DA INFORMAÇÃO



Julgue os itens subsecutivos, com relação ao tuning de banco de dados.

[1] No SQL Server, o uso de variáveis de tabela permite aumentar o desempenho de determinadas consultas.

25. ANO: 2013 BANCA: CESPE ÓRGÃO: CRPM PROVA: ANALISTA EM GEOCIÊNCIAS - SISTEMAS DE INFORMAÇÃO

Com relação a técnicas de análise de desempenho e otimização de consultas SQL, julgue os itens seguintes.

[1] No SQL Server, o comando TRUNCATE TABLE remove todos os dados de uma tabela e, se a tabela tiver uma coluna de identidade, provoca a reinicialização do contador de identidade.

26. ANO: 2013 BANCA: CESPE ÓRGÃO: TCE-RO PROVA: AUDITOR DE CONTROLE EXTERNO - CIÊNCIAS DA COMPUTAÇÃO

Julgue os itens subsequentes, com base nos conceitos de modelagem relacional dos dados e de administração de dados.

[1] Os backups gerados por uma versão mais recente do SQL Server não podem ser restaurados com o uso de versões anteriores.

27. ANO: 2013 BANCA: CESPE ÓRGÃO: BACEN PROVA: ANALISTA DO BACEN - ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Considerando os sistemas de gerenciamento de bancos de dados, julgue o próximo item.

[1] PostgreSQL, MariaDB, IIS e SQL Server são exemplos de sistemas gerenciadores de bancos de dados.

28. ANO: 2012 BANCA: CESPE ÓRGÃO: TJ-AC PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SISTEMAS

Com relação a banco de dados, julgue os itens que se seguem.

[1] No SQL SERVER 2012, permite-se a combinação da cláusula TOP com OFFSET e FETCH no mesmo escopo de consulta.

29. CEBRASPE (CESPE) - Analista Judiciário (TJ AC)/Técnico-Administrativa/Analista de Sistemas/2012

Com relação a banco de dados, julgue o item que se segue.

Quando a instrução SELECT DISTINCT é utilizada no SQL SERVER 2012, os nomes e aliases de coluna deverão ser definidos na lista de seleção.

30. ANO: 2012 BANCA: CESPE ÓRGÃO: TJ-AC PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SUPORTE

A respeito do banco de dados SQL Server e da linguagem SQL, julgue os itens seguintes.



[1] O comando `SELECT GETDATE()` recebe como retorno a data e a hora atual do sistema operacional. Para se retornar um usuário qualquer conectado a uma base de dados, deve-se executar o comando `SELECT SYSTEM_ADMIN`.

31. ANO: 2012 BANCA: CESPE ÓRGÃO: TJ-AC PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SUPORTE

A respeito do banco de dados SQL Server e da linguagem SQL, julgue os itens seguintes.

[1] Ao se executar o comando `SELECT ROUND(2.1234,1) AS Valor1`, obtém-se Valor1 2.1000 como resultado.



GABARITO

1. E
2. C
3. E
4. E
5. C
6. C
7. E
8. C
9. C
10. C
11. E
12. C
13. C
14. C
15. C
16. E
17. A
18. D
19. C
20. C
21. D
22. E
23. C
24. E
25. C
26. C
27. E
28. E
29. C
30. E
31. C



LISTA DE QUESTÕES - MULTIBANCAS

1. ISS São José dos Campos/FGV/2024

Com relação aos spinlock no MS SQL Server 2019, avalie as afirmativas a seguir.

- I. Os spinlocks não são uma exclusividade do MS SQL Server. São utilizados pelo sistema operacional quando é necessário obter acesso a uma determinada estrutura de dados somente por um curto intervalo de tempo. Por exemplo, quando uma thread não consegue obter acesso ao tentar adquirir um spinlock, ela é executada de maneira periódica se o recurso está disponível ou não, em vez de fazer uma suspensão imediata. Após algum tempo, a thread que está aguardando o spinlock será suspenso antes de poder adquirir o recurso. A suspensão permite que outras threads em execução na mesma CPU sejam executadas.
- II. O MS SQL Server usa spinlocks para proteger o acesso a algumas estruturas internas de dados. Os spinlocks são usados no mecanismo para serializar o acesso a determinadas estruturas de dados de maneira semelhante aos locks.
- III. A principal diferença entre o lock e o spinlock é o fato de que os locks executar um loop por um período verificando a disponibilidade de uma estrutura de dados. Por outro lado, uma thread tentando obter acesso a uma estrutura protegida por spinlock cederá imediatamente se recurso não estiver disponível. A suspensão requer a alternância de contexto de uma thread fora da CPU para que outra thread possa ser executada.

Está correto o que se afirma em

- a) III, apenas.
- b) II e III.
- c) II, apenas.
- d) I e II.
- e) I e III.

2. Câmara Municipal de São Paulo/FGV/2024

Com relação ao Microsoft SQL Server Profiler, avalie se as afirmativas a seguir são falsas (F) ou verdadeiras (V).

- I. É uma interface gráfica do usuário para o SQL Trace para monitorar uma instância do SGBD ou do Analysis Services. Permite que o analista monitore um ambiente de produção para ver quais procedimentos armazenados estão afetando o desempenho por serem executados com muita lentidão.
- II. Permite que o analista capture uma série de instruções Transact-SQL que levam a um problema de desempenho. O rastreamento salvo pode então ser usado para replicar o problema em um servidor de teste onde o problema pode ser diagnosticado.
- III. Oferece suporte à auditoria das ações executadas em instâncias do SGBD SQL Server. As auditorias registram ações relacionadas à segurança para revisão posterior por um administrador de segurança.

As afirmativas são, respectivamente,

- a) V-V-V.



e) F–F–F.

3. Câmara Municipal de São Paulo/FGV/2024

Os processos de otimização de consultas SQL em ambientes Oracle podem utilizar diversos tipos de índices.

O tipo de índice adequado para colunas que possuem um número relativamente baixo de valores, para aplicações de armazenamento de dados em que há baixa atividade de declarações SQL do tipo DML e filtragens ad hoc e eficiente em consultas com a função COUNT () denomina-se índice

- a) baseado em função.
- b) chave reversa.
- c) particionado.
- d) árvore B.
- e) bitmap.

4. FUNDATEC - Ana Sist (BRDE)/BRDE/Administração de Banco de Dados/2023

O DBA de um banco de dados SQL Server 2019 precisou procurar no catálogo algumas definições existentes sobre um esquema da base de dados específico para executar o seu trabalho. Ele descobriu que:

- Para procurar as tabelas existentes, deveria consultar a SYS.TABLES, e para encontrar as chaves primárias existentes, poderia consultar a tabela _____;
- Encontraria na tabela _____ o identificador do tipo de dado associado aos atributos de uma dada tabela;
- Para encontrar os procedimentos armazenados na base de dados, precisava consultar a SYS.PROCEDURES, e que se quisesse saber detalhes dos parâmetros de um dado procedimento armazenado, poderia encontrá-los na _____;

Assinale a alternativa que preenche, correta e respectivamente, as lacunas do trecho acima.

- a) SYS.INDEXES – SYS.COLUMNNS – SYS.PARAMETERS
- b) SYS. INDEXES _KEYS – SYS.TYPES – SYS.PARAMETERS
- c) SYS.INDEXES – SYS. COLUMNNS – SYS.PROCEDURES_EXTENDED
- d) SYS.INDEXES – SYS.TYPES – SYS.PARAMETERS
- e) SYS.PRIMARY_KEYS – SYS.TYPES – SYS.PROCEDURES_EXTENDED



Considere que o DBA de um banco de dados SQL Server 2019 definiu o índice abaixo sobre a tabela TAB1:

```
CREATE NONCLUSTERED INDEX IDX1  
ON TAB1 (codcomp, dataInicial)  
WHERE DataFinal IS NOT NULL;
```

Sobre isso, analise as assertivas abaixo:

I. Esse comando cria um índice filtrado.

II. O índice criado por esse comando contribui à melhoria do desempenho de consultas principalmente quando a condição (*DataFinal IS NOT NULL*) é observada em um grande número de tuplas dessa tabela.

III. O formato de armazenamento primário do índice criado por esse comando é columnstore.

Quais estão corretas?

- a) Apenas I.
- b) Apenas II.
- c) Apenas III.
- d) Apenas II e III.
- e) I, II e III.

6. FUNDATEC - Ana Sist (BRDE)/BRDE/Administração de Banco de Dados/2023

Considere as seguintes assertivas sobre recursos de backup no SQL Server 2019, assinalando V, se verdadeiras, ou F, se falsas.

() Oferece diferentes modelos de recuperação, entre eles o total, diferencial e log de transações.



() Um backup diferencial captura apenas as extensões dos dados alterados desde o último backup diferencial.

() Se a base de dados foi criada com múltiplos arquivos de dados, é possível criar backups de arquivos individuais e restaurá-los individualmente.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- a) V – F – V.
- b) V – V – F.
- c) F – F – V.
- d) F – V – V.
- e) F – F – F.

7. FUNDATEC - ANC (PROCERGS)/PROCERGS/Suporte/Bancos de Dados/2023

Analise as assertivas abaixo sobre o sistema de banco de dados Microsoft SQL Server 2019 e assinale a alternativa correta.

I. Durante operações de JOIN, valores nulos (vazios) não são considerados uma combinação.

II. Os três tipos de JOIN são INNER, OUTER e CROSS.

III. Colunas usadas em uma condição de JOIN precisam ser do mesmo tipo.

- a) Todas estão corretas.
- b) Todas estão incorretas.
- c) Apenas I está correta.
- d) Apenas II está correta.
- e) Apenas III está correta.



8. FUNDATEC - ANC (PROCERGS)/PROCERGS/Suporte/Bancos de Dados/2023

Assinale a alternativa que apresenta um caso de uso que é mais adequado para uso do Accelerated Database Recovery (ADR) do Microsoft SQL Server 2019.

- a) Um sistema operando com transações de longa duração.
- b) Quando o banco de dados está operando com mirroring (espelhamento).
- c) Em um sistema que executa muitas atualizações pequenas, mas não agrupadas.
- d) Quando o sistema não tem sofrido com indisponibilidade.
- e) Em sistemas onde o espaço de armazenamento é mais importante do que disponibilidade.

9. (VUNESP - ATCE (TCM SP)/TCM SP/Técnico de Informática/2023)

O comando do Transact-SQL para criar um *stored procedure*, denominado Eight, passando o parâmetro Seven, para executar uma operação de seleção é:

- a) CREATE PROC @Eight #Seven AS
SELECT ...;
- b) CREATE PROC &Eight \$Seven AS
SELECT ...;
- c) CREATE PROC Seven.Eight AS
SELECT ...;
- d) CREATE PROC #Seven OF Eight AS
SELECT ...;
- e) CREATE PROC Eight @Seven AS
SELECT ...;

10. (VUNESP - ATCE (TCM SP)/TCM SP/Técnico de Informática/2023)

Considere o seguinte comando para a criação de um *trigger* no Transact-SQL:

```
CREATE TRIGGER Yellow ON Red  
AFTER UPDATE
```



Esse comando cria um *trigger* denominado

- a) Yellow, que é acionado quando algum registro da tabela Red é alterado, gerando uma mensagem de aviso.
- b) Yellow, que é acionado quando algum registro do banco de dados Red é excluído, gerando uma interrupção com um aviso.
- c) Red, que é acionado quando algum registro da tabela Yellow é alterado, enviando um *e-mail* ao responsável.
- d) Red, que é acionado quando alguma tabela do banco de dados Yellow é modificado, enviando um *e-mail* ao responsável.
- e) Red, que é acionado quando alguma chave primária da tabela Yellow for alterada, provocando uma parada do banco de dados.

11. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

O sistema gerenciador de bancos de dados Microsoft SQL Server 2012 possui a opção OFFSET que pode ser inserida em um comando SELECT. Tal opção especifica

- a) o número máximo de linhas que devem ser exibidas como resultado da execução do comando.
- b) o número de linhas a serem excluídas da tabela alvo, antes da exibição das linhas de resultado.
- c) o tempo mínimo para que a resposta solicitada seja exibida.
- d) o número máximo de usuários simultâneos no banco de dados em questão.
- e) quantas linhas devem ser ignoradas antes que o comando inicie a exibição das linhas de resultado.

12. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

O comando do sistema gerenciador de bancos de dados Microsoft SQL Server que desfaz as transações até um *savepoint* especificado é:

- a) ROLLBACK TRAN @<nome do savepoint>;
- b) ROLLBACK TRAN <nome do savepoint>;
- c) ROLLBACK TRAN UNTIL <nome do savepoint>;
- d) ROLLBACK TRANSACTION FOR <nome do savepoint>;



Considere o seguinte comando do sistema gerenciador de bancos de dados Microsoft SQL Server:

TRUNCATE TABLE Hand;

O resultado da execução desse comando é

- a) excluir metade das tabelas do banco de dados Hand.
- b) excluir todos os registros da tabela Hand.
- c) excluir metade dos registros da tabela Hand.
- d) limitar o tamanho de todos os registros da tabela Hand.
- e) limitar o número de tabelas do banco de dados Hand.

14. (VUNESP - ACE (TCM SP)/TCM SP/Tecnologia da Informação/2023)

A linguagem Transact SQL do sistema gerenciador de bancos de dados Microsoft SQL Server possui o comando BULK INSERT, cuja função é

- a) codificar os dados de uma tabela especificada nesse comando.
- b) confirmar as operações de inserção de dados realizadas em determinado período de tempo.
- c) importar um arquivo com dados para uma tabela ou visão da base de dados.
- d) restituir os registros excluídos de determinada tabela da base de dados.
- e) eliminar os registros que contenham valores nulos nas tabelas do banco de dados.

15. (VUNESP - Ana Sis (CM SBO)/CM SBO/2023)

O sistema gerenciador de banco de dados Microsoft SQL Server 2008 R2 permite replicar dados, considerando servidores e clientes. Dessa forma, um dos três tipos de replicação oferecidos por esse gerenciador denomina-se

- a) Parcial.
- b) Definida.
- c) Transacional.
- d) Permanente.
- e) Proporcional.

16. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023



A unidade mais comum de medida de área é o hectare.

17. CEBRASPE (CESPE) - AFM (Pref Fortaleza)/Pref Fortaleza/Geografia/2023

Acerca do cálculo de áreas com base no SQL Server, julgue o item a seguir.

A função ST_Area permite calcular a área de polígonos.

18. Ano: 2018 Banca: FCC Órgão: DPE-AM Cargo: Analista Área: Banco de Sistemas Questão: 47.

Um Analista de Sistemas deseja alterar a coluna quantidade, que faz parte da tabela pedido do banco de dados empresa, do tipo INT para o tipo DECIMAL(7,2). Para isso, utilizando Transact-SQL no SQL Server, deverá usar o comando

- MODIFY COLUMN quantidade TO DECIMAL (7, 2) FROM empresa;
- ALTER TABLE empresa ALTER COLUMN quantidade DECIMAL (7, 2);
- ALTER TABLE empresa SET quantidade TO DECIMAL (7, 2);
- MODIFY TABLE empresa SET COLUMN quantidade DECIMAL (7, 2);
- ALTER TABLE empresa MODIFY COLUMN quantidade TO DECIMAL (7, 2);

19. CESPE - Especialista Técnico (BNB)/Analista de Sistema/2018

Acerca de bancos de dados, julgue o item que se segue.

O código a seguir, criado no SQL Server 2017, apresenta uma visão materializada, especificamente devido ao argumento SCHEMABINDING.

```
CREATE VIEW VwTeste  
WITH SCHEMABINDING  
AS  
SELECT campo1 FROM tabela WHERE campo1 > 17;
```

Certo

Errado

20. FCC - Agente de Fiscalização à Regulação de Transporte (ARTESP)/Tecnologia da Informação/2017

Em relação à remoção de linhas no SQL Server, é correto afirmar:

- Uma tabela que tenha todas as linhas removidas permanece no banco de dados. A instrução DELETE só exclui linhas da tabela; a tabela deve ser removida do banco de dados usando a instrução DROP TABLE.

15. A instrução DELETE, ao ser utilizada com o argumento WHERE não faz distinção entre



- c) Diferentemente da instrução DELETE, uma tabela esvaziada usando a instrução TRUNCATE TABLE é removida do banco de dados, junto com seus índices e outros objetos associados.
- d) Pode-se usar a cláusula UP (n) para limitar o número de linhas que são excluídas em uma instrução DELETE. Neste caso a operação de exclusão é executada em uma seleção aleatória de n linhas.
- e) Caso seja utilizada a cláusula UP junto com TRUNCATE para excluir linhas em uma determinada ordem, será preciso usar UP junto com ORDER BY em uma instrução de subseleção.

21. FCC - Analista Judiciário (TST)/Apoio Especializado/Análise de Sistemas/2017

Um Analista de Sistemas deseja fazer um backup completo de um banco de dados SQL Server chamado vendas para um disco cujo caminho é definido por 'Z:\servidor_backup\vendas.bak', formatando a mídia e comprimindo o banco de dados, utilizando Transact-SQL. Para isso, terá que utilizar o comando

- a) `RMAN DATABASE vendas to 'Z:\servidor_backup\vendas.bak' WITH FORMAT, COMPRESSION;`
- b) `BACKUP DATABASE vendas TO DISK = 'Z:\servidor_backup\vendas.bak' WITH FORMAT, COMPRESSION;`
- c) `BACKUP FROM vendas TO Z:\servidor_backup\vendas.bak SET FORMAT, COMPRESSION;`
- d) `BACKUP vendas TO Z:\servidor_backup\vendas.bak -F, -C;`
- e) `BACKUP FULL vendas TO DISK = 'Z:\servidor_backup\vendas.bak' WITH CONSTRAINT='FORMAT, COMPRESSION';`

22. CESPE - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Suporte/2016

Acerca da configuração e administração dos bancos de dados SQL Server 2008 R2 e MySQL 5.7, julgue o item subsequente.

A ferramenta SQL Server Configuration Manager permite realizar configurações de modo que uma instância do SQL Server se inicie automaticamente quando o servidor for ligado.

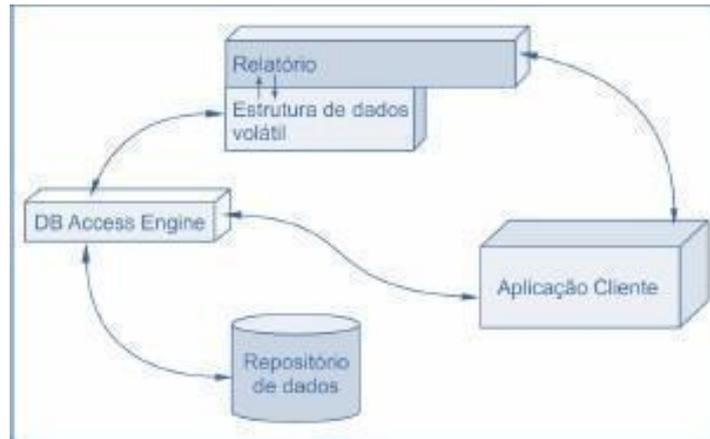
Certo

Errado

23. FCC - Auditor de Controle Externo (TCM-GO)/Informática/2015

A figura abaixo apresenta o diagrama da relação entre os principais elementos de uma solução baseada em SQL Server Reporting Services.





É correto afirmar sobre os elementos do diagrama:

- a) Os Relatórios podem ser de diversos tipos como: de análise, detalhado, de clickthrough, sub-relatório, histórico, armazenado em cache, instantâneo e relatório modelo.
- b) As Aplicações Cliente são capazes de implementar, gerenciar e visualizar os Relatórios, sendo os principais o Report XML Manager, Visual Studio, aplicações .NET por meio do ReportRDLX, além do aplicativo Office-SSRS Builder.
- c) Os Relatórios são baseados na linguagem RDLX (Report Definition XML Language), que é uma representação de XHTML criada pela Microsoft.
- d) DB Access Engine são os diversos mecanismos possíveis para acessar os dados.
▪ Corresponde às fontes de dados de acesso ao SQL Server e ao Microsoft Access, únicos SGBDs acessíveis.
- e) Repositório de Dados são as fontes de dados das quais se extraem as informações para os Relatórios criados e acessados através da linguagem SSRS.

24. CESPE - Analista Judiciário (TJDFT)/Apoio
Especializado/Suporte em Tecnologia da Informação/2015

A respeito da configuração e da administração de sistemas gerenciadores de bancos de dados (SGBD) e de produtos a eles relacionados, julgue o item a seguir.

Apenas instalações autônomas de SQL Server permitem o uso de servidor de arquivos SMB como opção de armazenamento.

Certo

Errado

25. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

O comando do Transact SQL do Microsoft SQL Server 2008 para criar um sinônimo, atribuindo a denominação 'teste' à tabela 'primeiros_programas', do banco de dados 'primeiro_db' é

- a) SYNONYM teste EQUAL primeiro_db.primeiros_programas
- b) CREATE SYNONYM teste FOR primeiro_db.primeiros_programas



e) MAKE SYNONYM teste OF primeiro_db.primeiros_programas

26. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

O comando do Transact SQL do Microsoft SQL Server 2008 para atualizar o valor de 20 registros, da coluna 'taxa', da tabela 'Blue', dividindo por 2 o valor dessa coluna 'taxa' é

- a) UPDATE MOST(20) Blue MAKE taxa = taxa/2
- b) UPDATE UPPER(20) Blue HAVING taxa = taxa/2.
- c) UPDATE SUP(20) Blue WITH taxa = taxa/2
- d) UPDATE TOP(20) Blue SET taxa = taxa/2
- e) UPDATE FIRST(20) Blue PRINTING taxa = taxa/2

27. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2015

No sistema gerenciador de banco de dados Microsoft SQL Server 2008, por meio da função sys.dm_index_physical_stats é possível verificar

- a) o número total de tabelas do banco de dados.
- b) a quantidade máxima de usuários simultâneos no banco de dados.
- c) o número total de usuários do banco de dados.
- d) a quantidade máxima permitida de tabelas no banco de dados.
- e) o nível de fragmentação de índices do banco de dados.

28. CESPE - Analista Administrativo (ANTAQ)/TI - Analista de Infraestrutura/2014

Acerca do Microsoft SQL Server 2008, julgue o seguinte item.

O argumento clustered do comando create index cria um índice em que a ordem lógica dos valores da chave determina a ordem física das linhas correspondentes em uma tabela.

Certo

Errado

29. FCC - Agente de Defensoria Pública (DPE SP)/Administrador de Banco de Dados/2013

Um recurso presente no MS-SQL Server 2008 modela as entidades relacionadas ao SQL Server de uma organização em uma exibição unificada. O Gerenciador do Utilitário e os pontos de vista deste recurso no SQL Server Management Studio fornecem aos administradores uma exibição holística da integridade dos recursos do SQL Server por meio de uma instância do SQL Server que funciona como um UCP (ponto de controle do utilitário). As entidades que podem ser exibidas no SQL Server UCP incluem:

- Instâncias do SQL Server.
- Aplicativos da camada de dados.
- Arquivos de banco de dados.



- a) SQL Entity Manager.
- b) UCP Manager.
- c) SQL Server Utility.
- d) SQL Query Administrator.
- e) Server Control Panel.

30. FCC - Agente de Defensoria Pública (DPE SP)/Analista de Sistemas/2013

Uma das funções disponíveis no sistema gerenciador de bancos de dados Microsoft SQL Server 2008 é @@CONNECTIONS. O resultado da aplicação dessa função é a obtenção

- a) do número de acessos autorizados, desde quando o SQL Server foi iniciado pela última vez.
- b) do número de tentativas de conexão bem ou mal sucedidas, desde quando o SQL Server foi iniciado pela última vez.
- c) do número de usuários simultâneos conectados ao SQL Server, no momento da execução dessa função.
- d) do tempo total de conexão de todos os usuários simultâneos do SQL Server, no momento da execução dessa função.
- e) da velocidade de conexão do usuário corrente ao Microsoft SQL Server.

31. CESPE - Analista do Banco Central do Brasil/Área 5 - Infraestrutura e Logística/2013

A respeito do SQL Server 2012 Reporting Services e do Microsoft Office 2010, julgue o item subsequente.

Power View é um recurso do Suplemento SQL Server 2012 Reporting Services para Microsoft SharePoint Server 2010 Enterprise Edition. Esse recurso permite criar relatórios ad hoc intuitivos para usuários comerciais, além de criar e interagir com exibição de dados de modelos de dados com base em pastas de trabalho PowerPivot.

Certo

Errado

32. CESPE - Analista Administrativo (TCE-ES)/Informática/2013

A ferramenta presente no SQL Server que, após analisar uma carga de trabalho, pode recomendar a adição, remoção ou modificação de estruturas de design físicas em bancos de dados é denominada

- a) SQL Server Studio.
- b) SQL Server Admin.
- c) SQL Server Configuration Manager.
- d) Orientador de Otimização do Mecanismo de Banco de Dados.
- e) SQL Server Profiler.



tabelas no banco de dados MeuBanco.

```
CREATE TABLE Tab1
  (Codigo int PRIMARY KEY NOT NULL,
   Descricao text NULL);

CREATE TABLE Tab2
  (Cod int PRIMARY KEY NOT NULL,
   Descricao text NULL,
   Fonte int NOT NULL,
   CONSTRAINT FK_Tab2_Fonte FOREIGN KEY
  (Fonte) REFERENCES Tab1 (Codigo));

CREATE TABLE Tab3
  (ID_Seq int PRIMARY KEY NOT NULL,
   Inscricao int UNIQUE NOT NULL,
   Responsavel int NULL,
   CONSTRAINT FK_Tab3_Responsavel FOREIGN KEY
  (Responsavel) REFERENCES Tab3 (Inscricao));
```

A figura abaixo representa o conteúdo das tabelas Tab1, Tab2 e Tab3 de MeuBanco.

Tab1

| Codigo | Descricao |
|--------|-----------|
| 10 | Alfa |
| 20 | Beta |
| 30 | Qui |

Tab2

| Cod | Descricao | Fonte |
|------|-----------|-------|
| 1000 | Zeta | 30 |
| 2000 | Psi | 10 |
| 3000 | Xi | 20 |
| 4000 | Omega | 10 |
| 5000 | Upsilon | 20 |
| 6000 | Tau | 30 |

Tab3

| ID_Seq | Inscricao | Responsavel |
|--------|-----------|-------------|
| 1 | 201601 | 201603 |
| 2 | 201602 | 201602 |
| 3 | 201603 | 201603 |
| 4 | 201604 | 201604 |
| 5 | 201605 | 201603 |
| 6 | 201606 | 201602 |

Em momento posterior, os comandos abaixo foram executados individualmente na seguinte ordem:

```
TRUNCATE TABLE Tab1;
```



Considere a execução de commit implícitos e desconsidere quaisquer comandos reconhecidos unicamente por aplicativos clientes para acesso aos bancos de dados do SQL Server 2012. Após a execução dos comandos, é correto afirmar que:

- as linhas de Tab1 foram removidas e foi registrada uma entrada no log de transações para cada linha excluída;
- as linhas, a estrutura da tabela, as colunas e as constraints de Tab1 forma removidas;
- Tab2 ficou vazia, mas sua estrutura não sofreu alterações;
- os dados de Tab3 foram removidos e suas respectivas páginas de dados foram desalocadas;
- Tab3 teve seus dados removidos e foi registrada uma entrada no log de transações para cada linha excluída.

34. BANCA: VUNESP ANO: 2012 ÓRGÃO: TJ-SP PROVA: ANALISTA JUDICIÁRIO -
COMUNICAÇÃO E PROCESSAMENTO DE DADOS

No Microsoft SQL Server 2008, as teclas de atalho para inserir ou remover um bookmark são:

- A Ctrl + K
- B Ctrl + B
- C Alt + U
- D Alt + K
- E Alt + B

35. BANCA: VUNESP ANO: 2013 ÓRGÃO: IMESC PROVA:
ANALISTA DE TECNOLOGIA - INFORMÁTICA

O sistema gerenciador de bancos de dados Microsoft SQL Server 2008 conta com um recurso denominado IntelliSense que provê facilidades para a edição de comandos. O botão do SQL Server 2008 que permite habilitar ou desabilitar esse recurso é

- A 
- B 
- C 
- D 
- E 



COMUNICAÇÃO E PROCESSAMENTO DE DADOS

Um usuário do Microsoft SQL Server 2008 deseja atribuir um sinônimo de nome S1 para a tabela Produtos. O código do Transact SQL para executar essa função é

- A ATTRIB SYM S1 FOR Produtos;
- B ATTRIB SYNONIM Produtos (S1);
- C CREATE SYNONYM S1 FOR Produtos;
- D CREATE SYNM S1 (Produtos);
- E CREATE SYM S1 Produtos;

37. BANCA: VUNESP ANO: 2013 ÓRGÃO: CETESB PROVA: ANALISTA DE TECNOLOGIA DA INFORMAÇÃO - ADMINISTRADOR DE BANCO DE DADOS

O procedimento armazenado do Transact do SQL Microsoft SQL Server 2008 que exibe informações sobre as dependências entre objetos (tabelas, visões, ...) de um banco de dados é

- A sp_lock.
- B sp_fkeys.
- C sp_depends.
- D sp_enumdsn.
- E sp_grantlogin.

38. BANCA: VUNESP ANO: 2013 ÓRGÃO: CETESB PROVA: ANALISTA DE TECNOLOGIA DA INFORMAÇÃO - ADMINISTRADOR DE BANCO DE DADOS

Sobre os Sistemas Gerenciadores de Bancos de Dados (considerando o Microsoft SQL Server 2008), é correto afirmar que

- A devem ser desligados por, pelo menos, 2 horas diariamente.
- B devem tratar comandos emitidos pelo usuário, permitindo, por exemplo, a busca de dados.
- C não podem ser utilizados em ambiente de rede.
- D podem ser substituídos, sem perda de funcionalidade, por um programa de gerenciamento de tarefas.
- E podem funcionar sem a presença de um sistema operacional no servidor.

39. BANCA: VUNESP ANO: 2013 ÓRGÃO: COREN-SP PROVA: ANALISTA - ADMINISTRADOR DE BANCO DE DADOS

Considerando o tuning do sistema gerenciador de bancos de dados Microsoft SQL Server 2012, um dos parâmetros monitorados refere-se ao uso do disco rígido, que indica

A a porcentagem de tempo de ocupação do disco com operações de leitura e escrita.



D o número de acessos por minuto ao disco rígido.

E o número médio de bytes lidos e escritos em um segundo.

40. BANCA: VUNESP ANO: 2014 ÓRGÃO: PRODEST-ES PROVA: ANALISTA DE TECNOLOGIA DA INFORMAÇÃO - DESENVOLVIMENTO DE SISTEMAS

No Transact SQL do sistema gerenciador de bancos de dados MS SQL Server 2008 R2, o comando geral para a criação de um procedimento armazenado é:

A CREATE PROCEDURE <nome do procedimento>

...

AS

<comando

SQL> GO

B FORM PROCEDURE <nome do procedimento>

...

OF

<comando

SQL> GO

C CREATE PROCEDURE <nome do procedimento>

...

LIKE

<comando SQL>

GO

D GENERATE PROCEDURE <nome do procedimento>

...

HOW

<comando SQL>

GO

E MAKE PROCEDURE <nome do procedimento>

...

OF TYPE

<comando SQL>

GO

41. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE TÉCNICO - DESENVOLVEDOR

O comando do sistema gerenciador de bancos de dados Microsoft SQL Server 2008 que possibilita remover o acesso de um usuário é:

A DELETE MEN <nome>



D DROP ACTION <nome>
E EXCEPT ACTION
<nome>

42. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE TÉCNICO - DESENVOLVEDOR

No sistema gerenciador de bancos de dados Microsoft SQL Server 2008, o comando SAVE TRANSACTION tem a função de

- A criar um trigger.
- B definir um ponto de salvamento de transações.
- C encerrar todas as conexões com o banco de dados.
- D fazer o backup de um banco de dados.
- E fazer o backup de uma tabela.

43. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

Uma forma de inserir comentários em um comando SQL no sistema gerenciador de banco de dados Microsoft SQL Server 2012 é

- A @ texto do comentário
- B § texto do comentário
- C -- texto do comentário
- D % texto do comentário %
- E # texto do comentário #

44. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

No sistema gerenciador de banco de dados Microsoft SQL Server 2008, o procedimento armazenado que exibe informações sobre os usuários atuais do banco de dados é

- A sp_who
- B sp_lock
- C sp_pkeys
- D sp_catalogs
- E sp_grantlogin.

45. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS



- A de tabelas com valores nulos.
- B de transações ativas na conexão atual.
- C de triggers disparados durante a conexão atual.
- D de usuários conectados ao servidor.
- E médio de registros por tabela.

46. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

O comando do sistema gerenciador de banco de dados Microsoft SQL Server 2000 para parar imediatamente o gerenciador é

- A KILL
- B STOP
- C LOCK
- D REVOKE
- E SHUTDOWN

47. BANCA: VUNESP ANO: 2013 ÓRGÃO: MPE-ES PROVA: AGENTE ESPECIALIZADO - ANALISTA DE BANCO DE DADOS

O comando do sistema gerenciador de banco de dados Microsoft SQL Server 2000 que tem como retorno o número de leituras de disco, desde a última ativação do gerenciador é

- A SELECT @@TOTAL_READ
- B SELECT @@PROCID
- C SELECT @@DBTS
- D SELECT @@IDLE
- E SELECT @@SPID



GABARITO

1. D
2. A
3. E
4. A
5. A
6. C
7. C
8. A
9. E
10. A
11. E
12. ANULADA
13. B
14. C
15. C
16. E
17. C
18. B
19. E
20. A
21. B
22. C
23. A
24. E
25. B
26. D
27. E
28. C
29. C
30. B
31. C
32. D
33. D
34. A
35. D
36. C
37. C
38. B
39. A



42. B

43. C

44. A

45. B

46. E

47. A



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.