

Formação de DBAs SQL Server 2008

Parte 2: Instalação e Configuração

Material Cedido pelo Professor Leandro Coelho Correia

Planejamento da Instalação



- Avaliação de requisitos:
 - Configuração mínima do hardware: CPU, memória, espaço em disco, formas de armazenamento;
 - Arquitetura da processador (32 bits x 64 bits): x86, x64, I64;
 - Necessidade de instalação do IIS (Internet Information Server) versão 5.0 ou superior quando o Reporting Service for utilizado;
 - Avaliação da versão do sistema operacional;
 - Necessidade de instalação do Windows Installer 3.1 ou superior;
 - Instalação do Framework .Net 2.0 será realizada automaticamente.

Planejamento da Instalação



- Definição de quais componentes serão instalados:
 - SQL Server Database Server;
 - Analysis Services;
 - Reporting Services;
 - Notification Services;
 - Integration Services;
 - Workstations components, books online, and development tools.

Planejamento da Instalação



- Contas utilizadas para executar os serviços:
 - **Local Service Account:** Conta com privilégios limitados e acesso a recursos de rede com credenciais nulas;
 - **Local System Account:** Conta local altamente privilegiada que deve ser evitada para instalação do SQL Server e SQL Server Agent;
 - **Network Service Account:** Similar à Local Service Account, mas com acesso à rede usando as credenciais da máquina local. Deve ser evitada para instalação do SQL Server e SQL Server Agent;
 - **Domain User Account:** Conta de usuário do domínio com privilégios configuráveis.

Boas Práticas:

Utilize a conta com o menor privilégio possível e que atenda às necessidades de instalação

Planejamento da Instalação



- Tipos de instalação:
 - Side-by-side
 - Instala uma nova instância do SQL Server permitindo a convivência de diferentes versões na mesma máquina.
 - Upgrade
 - Atualização uma instância já instalada para a versão 2008 do SQL Server.
- Modalidade de licenciamento:
 - User client access licensing
 - Uma licença para o servidor e outra para cada usuário acessando o servidor.
 - Device client access licensing
 - Uma licença para o servidor e outra para cada dispositivo acessando o servidor.
 - Processor licensing
 - Uma licença para cada processador.

Planejamento da Instalação



- Definição do collation:
 - O collation é uma configuração que determina como os dados serão ordenados e comparados;
 - Existem dois tipos de collations:
 - SQL Server Collations: mantidos somente para compatibilidade com versões anteriores do SQL Server;
 - Windows Collations: deve ser escolhido caso não haja necessidade de compatibilidade com versões anteriores do SQL Server.

Boas Práticas:

Utilize o mesmo collation para todas as instalações do SQL Server de uma mesma empresa.

Planejamento da Instalação



- Modo de autenticação:
 - Define a forma como os usuários podem se autenticar no SQL Server 2008;
 - Existem dois modos de autenticação:
 - Windows Authentication Mode:
 - O SGBD utiliza as credenciais do sistema operacional para autenticar o usuário;
 - Se todos os usuários estiverem no mesmo domínio, este é o modo de autenticação recomendado.
 - Mixed Mode:
 - Permite autenticação através do sistema operacional e através de usuários criados no próprio SQL Server.

Configurando o SQL Server 2008



- sp_configure
 - Stored procedure utilizada para visualizar e alterar as configurações de uma instância do SQL Server;
 - Sintaxe:

```
sp_configure [ [ @configname = ] 'option_name'  
[ , [ @configvalue = ] 'value' ] ]
```
- RECONFIGURE
 - Comando responsável pela atualização da configuração corrente após a execução da procedure sp_configure;
 - Sintaxe:

```
RECONFIGURE [ WITH OVERRIDE ]
```

Configurando o SQL Server 2008



- Configurações necessárias após a instalação:
 - CPU (affinity mask);
 - Memória (dinâmica, estática, valores mínimo e máximo);
 - Rede (bibliotecas, protocolos, portas e tipos de acesso);
 - Configurações dos bancos de sistema (master, model, msdb e tempdb).

Configurações de CPU



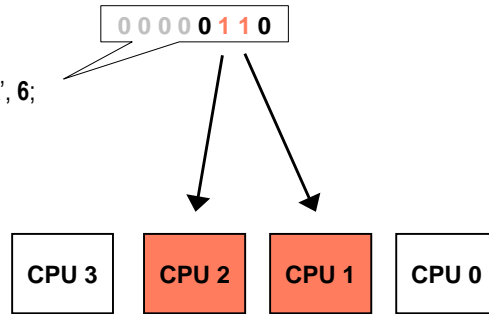
- O SQL Server 2008 oferece suporte a multiprocessamento:
 - Vários processadores;
 - Hyperthreading;
 - Multi-core.
- É possível configurar a quantidade de processadores disponíveis para o SQL Server através da propriedade *affinity mask*;
- É possível configurar a quantidade de processadores disponíveis ao processamento de operações de I/O do SQL Server, através da propriedade *affinity I/O mask*;
- Os valores das propriedades *affinity mask* e *affinity I/O mask* são convertidos em binário e definem quais as CPUs que o SQL Server irá utilizar;
- As configurações de *affinity mask* e *affinity I/O mask* devem ser sempre complementares e nunca conflitantes;
- A *affinity I/O mask* é uma propriedade avançada e deve ser configurada somente em situações específicas, para ambientes muito complexos.

Configurações de CPU



- Exemplo:
 - Máquina com 4 CPUs:

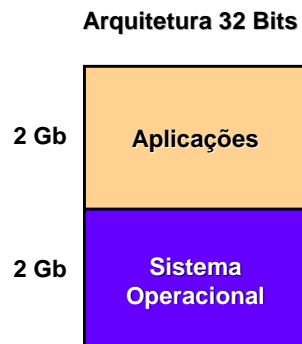
`sp_configure 'affinity mask', 6;
RECONFIGURE;`



Configurações de Memória



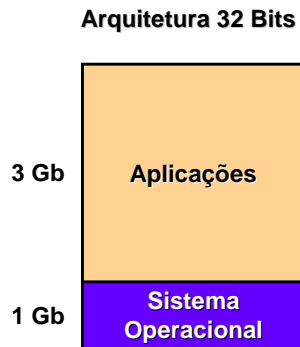
- Considerações gerais sobre memória:
 - Na arquitetura 32-bits o Windows reserva um espaço de endereçamento de até 2 Gb para as aplicações e o restante fica reservado ao sistema operacional.



Configurações de Memória



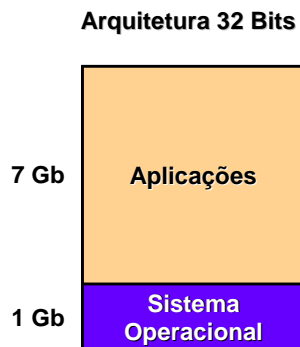
- Considerações gerais sobre memória:
 - Através do parâmetro de inicialização /3Gb, a área reservada para o sistema operacional passa a ser de 1Gb, deixando 3Gb para as aplicações.



Configurações de Memória



- Considerações gerais sobre memória:
 - Em máquinas com mais de 4Gb, o parâmetro de inicialização /PAE deve ser utilizado em conjunto com o /3Gb para permitir o máximo aproveitamento da memória.



Configurações de Memória



- No Windows 2003 o SQL Server 2008 trabalha por padrão com alocação e liberação dinâmica de memória, enquanto no Windows 2000 a opção padrão é alocação estática;
- Para poder utilizar a memória disponível em máquinas 32-bits com a opção /PAE configurada, é necessário habilitar a opção *AWE enabled* no SQL Server 2008;
- As propriedades *min server memory* e *max server memory* definem as quantidades mínima e máxima de memória que o SQL Server 2008 pode utilizar.

Configurações de Memória



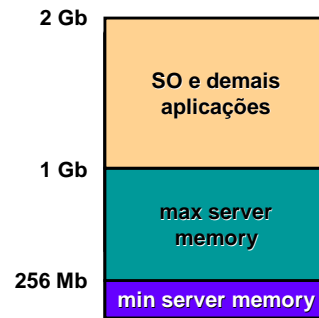
- Em servidores dedicados ao SQL Server 2008, as variáveis de controle da quantidade de memória devem ser mantidas com os valores default, garantindo maior flexibilidade ao SGBD;
- Em servidores onde o SQL Server 2008 compartilha a memória com outras aplicações as variáveis *min server memory* e *max server memory* devem ser configuradas para evitar a falta de memória para o SGBD ou a paginação excessiva das demais aplicações.

Configurações de Memória



- Exemplo:

```
exec sp_configure 'min server memory', '262144';  
exec sp_configure 'max server memory', '1048576';  
RECONFIGURE;
```



Boas Práticas:

Evite compartilhar o servidor de banco de dados com outros serviços.

Configurações de Rede



- O SQL Server 2008 possui uma arquitetura de comunicação baseada em 3 camadas:
 - Protocolos de comunicação (ODBC, OLEDB, etc);
 - SNI - SQL Network Interface (TCP/IP, Named Pipes, Shared Memory, VIA);
 - Hardware.

Configurações de Rede



- SNI – SQL Network Interface: Conjunto de bibliotecas de rede (*net-libraries*) que implementam os protocolos de comunicação suportados pelo SQL Server 2008:
 - Named Pipes
 - Protocolo padrão do SQL Server 2008;
 - Pode ser utilizado tanto para comunicação local (cliente e servidor executando na mesma máquina) quanto para comunicação remota (cliente e servidor executando em máquinas diferentes);
 - Bom desempenho para redes locais (LAN), mas não é recomendado para redes de longa distância (WAN) pelo de exigir uma forte interação entre o cliente e o servidor durante a comunicação;
 - Fácil de configurar e manter.

Configurações de Rede



- Shared Memory
 - Similar ao protocolo named pipes, sendo que a biblioteca é executada em modo kernel;
 - Permite somente comunicação local, restringindo sua utilização normalmente para troubleshooting.
- TCP/IP
 - Protocolo amplamente utilizado para diversas aplicações;
 - Suporta, de forma eficiente, comunicação em redes locais, redes de longa distância e Internet;
 - Permite um grande número de configurações não somente no SQL Server 2008, mas também no Windows;
 - Manutenção é mais complexa do que o protocolo named pipes.
- VIA (Virtual Interface Adapter)
 - Protocolo de alto desempenho, mas que exige um hardware específico, tornando sua utilização mais custosa do que os demais protocolos.

Configurações de Rede



- MDAC (Microsoft Data Access Components)
 - Conjunto de tecnologias (ADO, ODBC, OLEDB) agrupadas em um framework para simplificar e padronizar o desenvolvimento de aplicações que realizam acesso a dados de forma significativa;
 - Foi distribuído com o SQL Server e outros pacotes de ferramentas Microsoft até a versão 2.8 (a última a conter suporte a funções específicas do SQL Server) e atualmente é distribuído em conjunto com o sistema operacional.

Configurações de Rede



- SNAC (SQL Native Client)
 - Conjunto de tecnologias que compõem uma API similar ao MDAC, contendo funções específicas do SQL 2008:
 - Database mirroring;
 - Operações assíncronas;
 - Novos tipos de dados (XML, tipos de dados de alta capacidade de armazenamento, tipos de dados definidos pelo usuário);
 - Multiple Active Result Sets (MARS);
 - Query notifications;
 - Mudanças e expiração de senhas;
 - Snapshot isolation (com algumas exceções);
 - Características específicas de criptografia.
 - Evidencia uma tendência de generalização do MDAC, deixando o SNAC para implementações mais especializadas.



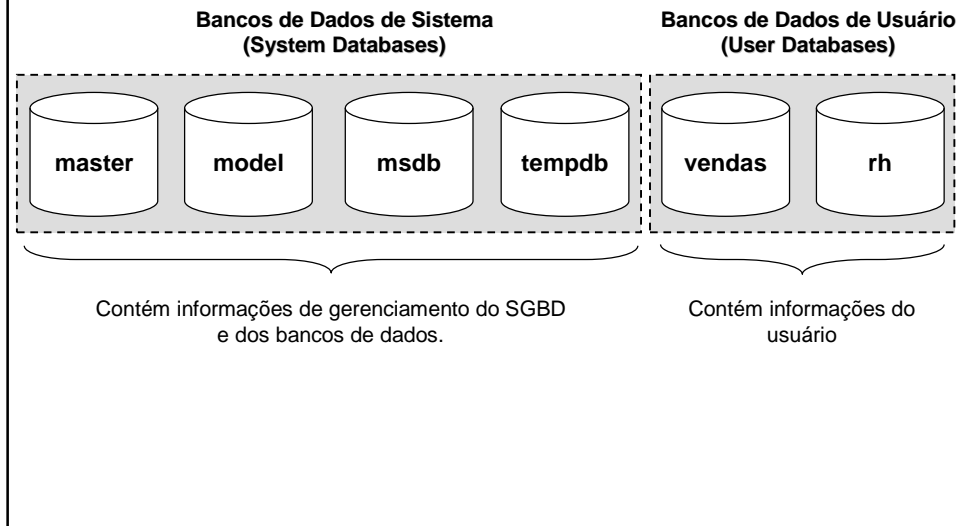
Criação e Configuração de Bancos de Dados

Banco de Dados no SQL Server 2008



- No SQL Server 2008 um banco de dados é um conjunto de objetos que contém dados (informações inseridas e manipuladas pelo usuário) e metadados (informações sobre a estrutura do banco de dados e seus objetos);
- Um banco de dados também contém objetos que implementam parte da lógica da aplicação (procedimentos armazenados e funções).

Bancos de Dados no SQL Server 2008



Bancos de Dados de Sistema



- master
 - Contém informações relativas a uma instância do SQL Server 2008 (contas de usuário; configurações de rede, memória e cpu; nomes dos bancos de dados e sua estrutura física; etc);
 - É o banco de dados mais importante para o correto funcionamento do SGBD;
 - Necessita de backup sempre que uma das operações abaixo for realizada:
 - Criação ou exclusão de banco de dados;
 - Mudança de configurações da instância;
 - Contas de usuário forem criadas ou alterada.

Bancos de Dados de Sistema



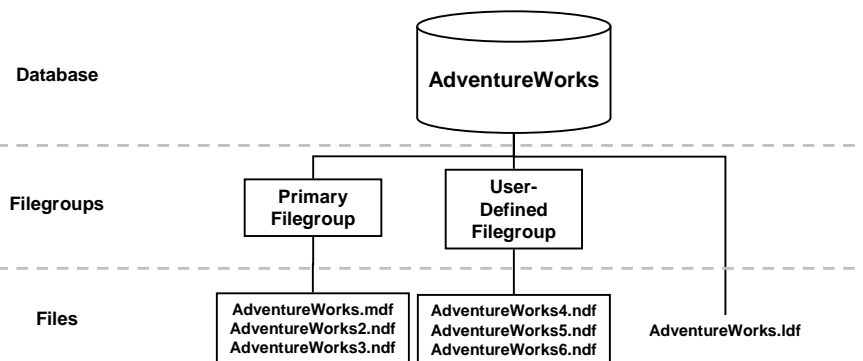
- model
 - Usado como modelo para a criação de novos bancos de dados;
 - Configurações como tamanho dos bancos de dados, modelo de recuperação e collation são aplicadas aos novos bancos de dados a partir do que estiver configurado no model.
- msdb
 - Contém informações sobre tarefas de manutenção periódicas como backup, planos de manutenção, jobs, alertas e operadores;
 - Serviços como o SQL Server Agent, Service Broker e Database Mail utilizam o msdb para armazenar suas configurações.
- tempdb
 - Contém informações não persistentes e é utilizado para armazenar objetos temporários como subconjuntos de dados, tabelas temporárias, etc.

Bancos de Dados de Sistema



- resource
 - Introduzido no SQL Server 2008 para simplificar processos de atualização de versão e rollbacks de service packs;
 - Banco de dados somente leitura e oculto que contém informações sobre procedimentos armazenados e funções;
 - Como não contém dados de usuário nem metadados, não precisa ser incluído em planos de backup.

Estrutura de um Banco de Dados



Estrutura de um Banco de Dados



- Files
 - São os arquivos físicos onde as informações dos bancos de dados estão armazenadas;
 - Todo banco de dados possui pelo menos dois arquivos:
 - Primary File (*.MDF): Arquivo que contém o catálogo do banco de dados;
 - Log File (*.LDF): Arquivo que armazena o log de transações.
 - Adicionalmente, um banco de dados pode conter outros arquivos de usuário (*.NDF).

Estrutura de um Banco de Dados



- Filegroups
 - São agrupamentos lógicos de arquivos;
 - Quando um objeto que armazena dados (tabelas e índices) é criado, o filegroup onde ele será armazenado deve ser especificado ou o objeto será criado no default filegroup;
 - Quando dados são inseridos em uma tabela ou índice, o SQL Server distribui os dados entre os arquivos do filegroup;
 - Todo banco de dados possui pelo menos um filegroup (Primary Filegroup) onde o primary file (*.MDF) fica armazenado;
 - Adicionalmente, outros filegroups podem ser criados;
 - O arquivo do log de transações (*.LDF) não está associado a nenhum filegroup.

Criando um Banco de Dados



Sintaxe básica:

```
CREATE DATABASE database_name
ON PRIMARY
( NAME = database_datafile_logical_name,
  FILENAME = database_datafile_physical_name,
  SIZE = database_datafile_initial_size,
  MAXSIZE = database_datafile_max_size,
  FILEGROWTH = database_datafile_growth )
LOG ON
( NAME = database_logfile_logical_name,
  FILENAME = database_logfile_logical_name,
  SIZE = database_logfile_logical_name,
  MAXSIZE = database_logfile_logical_name,
  FILEGROWTH = database_logfile_logical_name );
```

Criando um Banco de Dados



- NAME = Nome lógico do arquivo usado;
- FILENAME = Caminho e nome do arquivo no sistema operacional;
- SIZE = Tamanho inicial do banco de dados. Pode ser expresso em Kb, Mb (default), Gb ou Tb;
- MAXSIZE = Tamanho máximo que o banco de dados pode atingir. Pode ser expresso em Kb, Mb (default), Gb ou Tb;
- FILEGROWTH = Taxa de crescimento do banco de dados. Pode ser expressa em Kb, Mb (default), Gb, Tb ou %.

Criando um Banco de Dados



Exemplo 1: 1 database, 1 filegroup, 1 data file

```
CREATE DATABASE bd_treinamento
ON PRIMARY
( NAME = bd_treinamento_data,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento_data.mdf',
  SIZE = 50,
  MAXSIZE = 300,
  FILEGROWTH = 20 )
LOG ON
( NAME = bd_treinamento_log,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento_log.ldf',
  SIZE = 10,
  MAXSIZE = 100,
  FILEGROWTH = 10 );
```

Criando um Banco de Dados



Exemplo 2: 1 database, 1 filegroup, 2 data files

```
CREATE DATABASE bd_treinamento
ON PRIMARY
( NAME = bd_treinamento_data,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento_data.mdf',
  SIZE = 50,
  MAXSIZE = 300,
  FILEGROWTH = 20 ),
( NAME = bd_treinamento2_data,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento2_data.ndf',
  SIZE = 40,
  MAXSIZE = 200,
  FILEGROWTH = 10 )
LOG ON
( NAME = bd_treinamento_log,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento_log.ldf',
  SIZE = 10,
  MAXSIZE = 100,
  FILEGROWTH = 10);
```

Criando um Banco de Dados



Exemplo 3: 1 database, 2 filegroup, 4 data files

```
CREATE DATABASE bd_treinamento
ON PRIMARY
( NAME = bd_treinamento_data,
  FILENAME =
    'D:\MSSQL\DATA\bd_treinamento_data.mdf',
  SIZE = 50,
  MAXSIZE = 300,
  FILEGROWTH = 20 ),
( NAME = bd_treinamento2_data,
  FILENAME =
    'D:\MSSQL\DATA\bd_treinamento2_data.ndf',
  SIZE = 40,
  MAXSIZE = 200,
  FILEGROWTH = 10 )
```

```
ON FG2
( NAME = bd_treinamento3_data,
  FILENAME =
    'D:\MSSQL\DATA\bd_treinamento3_data.ndf',
  SIZE = 50,
  MAXSIZE = 300,
  FILEGROWTH = 20 ),
( NAME = bd_treinamento4_data,
  FILENAME =
    'D:\MSSQL\DATA\bd_treinamento4_data.ndf',
  SIZE = 40,
  MAXSIZE = 200,
  FILEGROWTH = 10 )
LOG ON
( NAME = bd_treinamento_log,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento_log.ldf',
  SIZE = 10,
  MAXSIZE = 100,
  FILEGROWTH = 10 )
);
```

Modificando um Banco de Dados



- Adicionando um novo arquivo ao banco de dados:

```
ALTER DATABASE database_name ADD FILE
( NAME = database_file_logical_name,
  FILENAME = database_file_physical_name,
  SIZE = database_file_initial_size,
  MAXSIZE = database_file_max_size,
  FILEGROWTH = database_file_growth ) TO FILEGROUP filegroup_name;
```

- Exemplo:

```
ALTER DATABASE bd_treinamento ADD FILE
( NAME = bd_treinamento5_data,
  FILENAME = 'D:\MSSQL\DATA\bd_treinamento5_data.ndf',
  SIZE = 80,
  MAXSIZE = 400,
  FILEGROWTH = 50 ) TO FILEGROUP FG2;
```

Modificando um Banco de Dados



- Modificando um arquivo existente:

```
ALTER DATABASE database_name MODIFY FILE
( NAME = database_file_logical_name,
  SIZE = database_file_initial_size,
  MAXSIZE = database_file_max_size,
  FILEGROWTH = database_file_growth );
```

- Exemplo:

```
ALTER DATABASE bd_treinamento MODIFY FILE
( NAME = bd_treinamento2_data, SIZE = 80 );

ALTER DATABASE bd_treinamento MODIFY FILE
( NAME = bd_treinamento2_data, FILEGROWTH = 0 );
```

Modificando um Banco de Dados



- Removendo um arquivo existente:

```
ALTER DATABASE database_name  
REMOVE FILE database_file_logical_name;
```

- Exemplo:

```
ALTER DATABASE bd_treinamento  
REMOVE FILE bd_treinamento5_data;
```

Modificando um Banco de Dados



- Adicionando um filegroup:

```
ALTER DATABASE database_name  
ADD FILEGROUP filegroup_name;
```

- Exemplo:

```
ALTER DATABASE bd_treinamento  
ADD FILEGROUP FG3;
```

Modificando um Banco de Dados



- Modificar o default filegroup:

```
ALTER DATABASE database_name  
MODIFY FILEGROUP filegroup_name DEFAULT;
```

- Exemplo:

```
ALTER DATABASE bd_treinamento  
MODIFY FILEGROUP FG2 DEFAULT;
```

Liberando Espaço do Banco de Dados



- DBCC SHRINKDATABASE
 - Libera espaço em disco disponível nas áreas de dados e log de um banco de dados até um percentual especificado;
 - Exemplo:
DBCC SHRINKDATABASE ('bd_treinamento', 20);
- DBCC SHRINKFILE
 - Libera espaço em disco disponível em um arquivo de dados ou log até um tamanho especificado;
 - Pode ser utilizado para esvaziar um arquivo de dados e permitir sua exclusão;
 - Exemplo:
DBCC SHRINKFILE ('bd_treinamento5_data', 40);
DBCC SHRINKFILE ('bd_treinamento5_data', EMPTYFILE);

Diretrizes para Criação de Bancos de Dados



- Performance:
 - Colocar o(s) arquivo(s) de log em discos separados daqueles que armazenam os arquivos de dados;
 - Criar os índices em filegroups dedicados, separando-os dos demais objetos (tabelas, visões, etc);
 - Utilizar discos organizados em RAID 5 para área de dados RAID 1 para área de log.
- Manutenção:
 - Dedicar o filegroup PRIMARY exclusivamente para o catálogo do banco (arquivo MDF), criando filegroups adicionais para os demais arquivos (NDF);
 - Criar filegroups com arquivos pequenos (até 2GB), adicionando novos arquivos quando necessário.

Configurando um Banco de Dados



- Algumas opções de configuração:
 - RECURSIVE_TRIGGERS
 - Permite habilitar/desabilitar a possibilidade de um trigger ser disparado de forma recursiva.
 - RECOVERY
 - Define o modelo de recuperação (recovery) a ser utilizado: SIMPLE, BULK_LOGGED ou FULL.
 - ONLINE / OFFLINE / EMERGENCY
 - Modifica o estado do banco de dados.

Configurando um Banco de Dados



- Mais opções de configuração:
 - SINGLE_USER / RESTRICTED_USER / MULTI_USER
 - Restringe o acesso ao banco de dados.
 - READ_ONLY / READ_WRITE
 - Define se operações de atualização serão permitidas no banco de dados.

Configurando um Banco de Dados



- Estados do banco de dados:
 - ONLINE
 - O banco de dados está consistente e disponível para utilização.
 - OFFLINE
 - O banco de dados está indisponível de forma a viabilizar manutenções como transferência de arquivos do banco de um disco para outro.
 - RESTORING
 - O banco de dados está indisponível em virtude de um processo de restore em andamento.
 - RECOVERING
 - O banco de dados está indisponível em virtude de um processo de recuperação (recovery) em andamento.

Configurando um Banco de Dados



- Estados do banco de dados:
 - RECOVERY PENDING
 - O processo de recuperação não pôde ser concluído pela falta de algum arquivo ou por falta de recursos no sistema. Uma intervenção do DBA é necessária para que o banco de dados retorne ao estado ONLINE.
 - SUSPECT
 - O filegroup primário está comprometido ou não foi encontrado durante o processo de recuperação. Uma intervenção do DBA é necessária para que o banco de dados retorne ao estado ONLINE.
 - EMERGENCY
 - Estado configurado pelo DBA para resolução de problemas no banco de dados. Somente administradores poderão acessar o banco de dados e o acesso será somente para leitura.

Configurando um Banco de Dados



- Restrições de acesso ao banco de dados:
 - SINGLE_USER
 - Somente um usuário poderá conectar ao banco de dados a cada momento;
 - Cláusulas de terminação do comando ALTER DATABASE permitem desconectar todos os usuários ativos de forma a permitir a configuração deste nível de acesso.
 - RESTRICTED_USER
 - Somente membros das roles db_owner, db_creator e sysadmin poderão ter acesso ao banco de dados;
 - Cláusulas de terminação do comando ALTER DATABASE permitem desconectar todos os usuários ativos de forma a permitir a configuração deste nível de acesso.
 - MULTI_USER
 - Qualquer usuário com permissão no banco de dados em questão poderá acessá-lo.

Configurando um Banco de Dados



- Configurações de atualização do banco de dados:
 - READ_ONLY
 - Somente operações de leitura poderão ser realizadas.
 - READ_WRITE
 - Operações de leitura e atualização poderão ser realizadas.

Configurando um Banco de Dados



- Exemplos de sintaxe dos comandos de configuração do banco de dados:

```
ALTER DATABASE database_name SET RECOVERY SIMPLE;
```

```
ALTER DATABASE database_name SET OFFLINE;
```

```
ALTER DATABASE database_name SET SINGLE_USER  
WITH ROLLBACK IMMEDIATE;
```

Excluindo um Banco de Dados



- **Sintaxe:**

```
DROP DATABASE database_name
```

- **Exemplo:**

```
DROP DATABASE bd_treinamento;
```

- O banco de dados e todos os seus arquivos serão excluídos;
- O usuário que está excluindo o banco de dados não pode estar no mesmo contexto do banco de dados a ser excluído.

Particionamento

Particionamento



- Definição:
 - Solução baseada em 2 características:
 - Divisão de um grande volume de dados em partes menores chamadas partições;
 - Um mecanismo de abstração para que as partições individuais sejam visualizadas pela aplicação como uma única partição.
- Objetivo do particionamento:
 - Aumento de performance;
 - Facilidade de administração.

Visões Particionadas

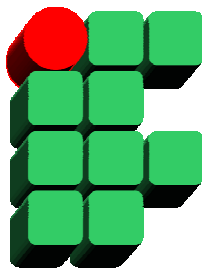


- Solução de particionamento disponibilizada a partir do SQL Server 2000;
- Composta de 4 etapas:
 - Definição de um critério de particionamento;
 - Criação e carga das tabelas individuais;
 - Criação de uma constraint de check no(s) campo(s) que define(m) o particionamento;
 - Criação de uma visão que reúna os dados das diversas tabelas.

Particionamento Físico



- Solução de particionamento oferecida pelo SQL Server 2008;
- Composta de 2 elementos:
 - Função de particionamento (Partition Function) que define as fronteiras das partições;
 - Distribuição física das partições (Partition Scheme) que determina onde cada partição será armazenada.



Formação de DBAs SQL Server 2008 Parte 2: Instalação e Configuração

Material Cedido pelo Professor Leandro Coelho Correia