

Apostila Microsoft SQL Server 7.0

Instalação, configuração e gerenciamento

1 - Introdução	4
Visão Geral do SQL Server	4
Sistemas Gerenciadores de Banco de Dados	6
2 - Instalação e Configuração	8
Requisitos de Sistema	8
Opções Usadas na Instalação	9
Instalando o software de servidor	12
Verificando se o SQL Server foi instalado corretamente	20
Instalando o software de cliente	24
Registrando um servidor	30
Solução de problemas de instalação	40
3 - Ferramentas de gerenciamento do SQL Server	43
SQL Server Enterprise Manager	44
SQL Server Client Network Utility	48
SQL Server Profiler	52
SQL Server Query Analyzer	54
SQL Server Books Online	56
Assistentes do SQL Server [SQL Server Wizards]	57
4 - Fundamentos de arquitetura do SQL Server	61
O Catálogo do Sistema	61
Componentes do Banco de Dados	64
Estrutura dos bancos de dados	65
5 - Criando Bancos de Dados	67
Gerenciando Bancos de Dados	67
Arquivos e grupos de arquivos	76
Criando Tabelas	79
Alterando a estrutura das tabelas	85
Definindo opções de bancos de dados	88
Considerações para melhor gerenciamento	91
Documentação dos passos de criação de bancos de dados no SQL Server	92
6 - Consultando Dados	95
A Sintaxe do SELECT	95
Manipulando expressões	98
Condições de pesquisa	103
Outros recursos	106
Alteração de Dados	108
Inserindo linhas	108
Excluindo linhas	110
8 - Consultas Avançadas	112
Dados de resumo	112
Junções de tabelas	115
Subconsultas	121
9 - Implementando Índices	Erro! Indicador não definido.
Por que índices?	Erro! Indicador não definido.
Tipos de Índices	Erro! Indicador não definido.

Características dos Índices _____	Erro! Indicador não definido.
Otimizando Consultas _____	Erro! Indicador não definido.
10 - Integridade de Dados _____	Erro! Indicador não definido.
A propriedade IDENTITY _____	Erro! Indicador não definido.
Usando Defaults e Regras _____	Erro! Indicador não definido.
Definindo e usando restrições [constraints] _____	Erro! Indicador não definido.
11 - Visões, Gatilhos e Procedimentos _____	Erro! Indicador não definido.
Visões [Views] _____	Erro! Indicador não definido.
Procedimentos Armazenados _____	Erro! Indicador não definido.
Gatilhos [Triggers] _____	Erro! Indicador não definido.
12 - Segurança _____	Erro! Indicador não definido.
Conceitos _____	Erro! Indicador não definido.
Papéis [Roles] _____	Erro! Indicador não definido.
Visualizando informações de segurança _____	Erro! Indicador não definido.
Permissões _____	Erro! Indicador não definido.
13 - Backup e Restauração _____	Erro! Indicador não definido.
Conceitos _____	Erro! Indicador não definido.
Dispositivos de Backup _____	Erro! Indicador não definido.
Implementando um backup _____	Erro! Indicador não definido.
Restaurando um Backup _____	Erro! Indicador não definido.
Agendando Backups Automáticos _____	Erro! Indicador não definido.
14 - Outros Recursos _____	Erro! Indicador não definido.
Configuração do SQL Server _____	Erro! Indicador não definido.
Entendendo as Tabelas de Sistema _____	Erro! Indicador não definido.
Importação e Exportação de Dados _____	Erro! Indicador não definido.
Publicando dados na Internet _____	Erro! Indicador não definido.
Modelo de exemplo _____	Erro! Indicador não definido.
Requisitos do Sistema _____	Erro! Indicador não definido.
Modelos de dados _____	Erro! Indicador não definido.
UPGRADE do SQL Server 6.x para a versão 7.0 _____	Erro! Indicador não definido.
Assistente de Atualização de Versão _____	Erro! Indicador não definido.

1 - Introdução

Visão geral do SQL Server

Sistemas Gerenciadores de Bancos de Dados

Objetivos:

- Obter uma visão geral do SQL Server e do seu funcionamento;
- Conhecer as ferramentas do SQL Server;
- Saber a divisão de papéis entre o administrador do sistema e o implementador de bancos de dados.

Visão Geral do SQL Server

O SQL Server é um sistema de gerenciamento de bancos de dados cliente/servidor de alto desempenho com alta integração com o Windows NT. Suas características são: Integração com os serviços de *multithreading* [múltiplas linhas], agendamento, Monitor de Desempenho, e log de eventos do Windows NT. Um usuário pode se conectar ao SQL Server com a mesma senha usada para a rede Windows NT.

Replicação nativa permite disseminar informações para vários locais, reduzindo a dependência de um servidor único, e deixando a informação necessária mais próxima de quem realmente precisa dela.

Arquitetura paralela, que executa as funções de banco de dados simultaneamente para diversos usuários e tira proveito de sistemas com múltiplos processadores.

Gerenciamento centralizado de todos os servidores através de uma arquitetura de gerenciamento distribuída, com uma interface visual de gerenciamento.

Distributed Management Framework (DMF)

O SQL Server possui uma arquitetura distribuída de gerenciamento [distributed management framework], composta de objetos, serviços e componentes. Através dela, vários servidores podem ser gerenciados completamente a partir de qualquer local na rede. Entre outros componentes, essa arquitetura é composta de:

- **SQL-DMO:** biblioteca de objetos ActiveX que expõe interfaces para todas as funções de gerenciamento do SQL Server e pode ser usada em qualquer linguagem compatível com automação ActiveX. Permite gerenciar servidores, bancos de dados, tabelas e outros objetos relacionados ao banco de dados.
- **SQL Enterprise Manager:** ferramenta gráfica de administração que, utilizando os objetos SQL-DMO, simplifica o gerenciamento de um ambiente de múltiplos servidores.
- **Serviços SQLServerAgent e MSSQLServer:** executando no servidor NT, o serviço SQLServerAgent permite agendar tarefas, como backups, por exemplo, e definir alertas para informar quando ocorrem condições de erro diversas. O serviço MSSQLServer é o

componente central, que permite inserir, atualizar e consultar dados armazenados no SQL Server.

Ferramentas de Administração

O SQL Server vem com várias ferramentas de administração que podem ser executadas a partir de um servidor Windows NT, de uma estação Windows NT Workstation, ou até mesmo a partir do Windows 95/98. São elas:



- *Enterprise Manager*: como já foi dito, gerencia vários servidores, permitindo executar qualquer tarefa relacionada ao SQL Server. Como será visto adiante, ele roda dentro MMC (Microsoft Management Console). Para executá-lo através de Iniciar, Executar, entre com a seguinte instrução:
mmc /s "pasta-base_do_SQLServer\BINN\SQL Server Enterprise Manager.MSC",
substituindo pasta-base_do_SQLServer pela pasta onde você instalou o SQL Server 7. Por padrão, é C:\MSSQL7.



- *Service Manager*(*SQLMANGR.EXE*): permite iniciar, pausar, continuar e parar ("finalizar") os serviços do SQL Server.



- *Query Analyzer*(*ISQLW.EXE*): permite administrar diretamente o SQL Server usando comandos Transact-SQL. Os comandos SQL podem ser executados interativamente, ou podem ser executados de procedimentos armazenados ou scripts.



- *Profiler* (*SQLTRACE.EXE*): permite monitorar toda a atividade do servidor e registrar essa atividade em arquivos de log, incluindo comandos SQL executados pelo servidor.



- *Client Network Utility* (*CLICONFG.EXE*): configura o software de acesso cliente numa estação.



- *Performance Monitor* (*SQLCTRS.PMC*: integra o Performance Monitor ("Desempenho do Sistema") do Windows NT com o SQL Server, para monitorar o desempenho do sistema.



- *Server Network Utility* (*SRVNETCN.EXE*): permite adicionar, remover ou configurar as Net-libraries, que são os protocolos aceitos para comunicação do cliente com o servidor.



- *SQL Server Books Online*: toda a documentação do SQL Server, para consultar on-line. Permite fazer pesquisas de texto na documentação. Para executá-lo, em Iniciar, Executar, entre com HH pasta_base_do_SQLServer\BOOKS\SQLBOL.CHM, onde

pasta_base_do_SQLServer é o diretório onde o SQL Server foi instalado. Por padrão, é C:\MSSQL7.



- *Uninstall SQL Server 7.0*: permite que você remova a instalação existente do SQL Server 7.0

Sistemas Gerenciadores de Banco de Dados

Um sistema gerenciador de banco de dados (SGBD) como o SQL Server é responsável por armazenar dados de forma confiável e permitir fácil recuperação e atualização desses dados. Um SGBD *relacional* armazena dados de forma relacional, isto é na forma de linhas e colunas.

Conceitos Relacionais

Um *registro* [record] ou *linha* [row] é um grupo de variáveis com tipos de dados diferentes, que armazenam dados relacionados. Por exemplo, um registro pode conter os dados relativos a um produto vendido pela empresa, como descrição, código de identificação, quantidade em estoque.

Um *campo* [field] ou *coluna* [column] é um dos itens de informação dentro de uma linha da tabela, como a descrição da informação.

Uma *tabela* [table] é um conjunto de linhas (registros) com a mesma estrutura, armazenados de forma permanente em disco. As tabelas são compostas de linhas (row) ou registros (record) e colunas (column) ou field (campo).

Um *banco de dados* [database] é um conjunto de tabelas que contêm dados relacionados. Por exemplo, um sistema de contas a pagar poderia ter um banco de dados de contas a pagar, com uma tabela para duplicatas, uma tabela para bancos, uma tabela para contas etc.

Um *índice* [index, plural 'indexes' ou 'indices'] é um mecanismo que permite pesquisar rapidamente por linhas em uma tabela, dado o valor de uma determinada coluna (ou algumas colunas) da tabela. Um *índice primário* ou *chave primária* define um valor único, que não pode ser repetido em outras linhas da tabela.

Uma *consulta* [query] é um pedido de pesquisa no banco de dados, que permite obter todo um subconjunto da tabela ou de várias tabelas, especificando as condições de seleção.

Desktop x Cliente/Servidor

Uma aplicação que utiliza bancos de dados é composta de três partes:

- *Interface com o usuário*: responsável por validar as entradas do usuário, e iniciar pesquisas de acordo com um pedido do usuário.
- *Mecanismo de acesso* [database engine]: responsável pela manutenção das estruturas de dados necessárias em arquivos, pelos detalhes internos do acesso aos dados, e pela manutenção da integridade dos dados.
- *Armazenamento de dados*: arquivos que contêm os dados em si.

Um banco de dados "desktop" (ou baseado em arquivos) é aquele no qual a interface com o usuário e o mecanismo de acesso ficam no mesmo computador (a estação) e

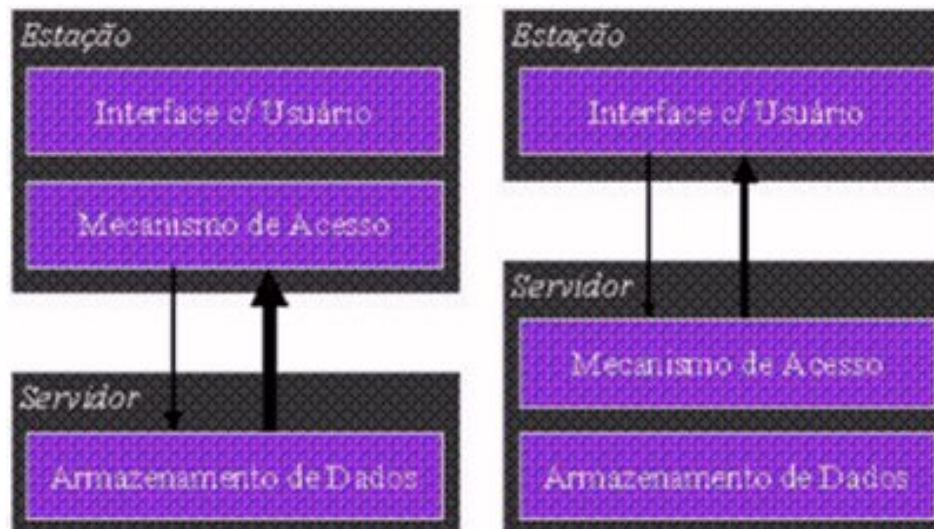
apenas os arquivos de dados ficam num servidor de rede. Operações de consulta ou pesquisa devem passar através da rede.

Por exemplo, quando um usuário quer ver uma relação de contas a pagar, mas apenas em determinado período, o sistema deve selecionar alguns registros baseado na data informada. No ambiente desktop, a estação traz todos os registros através da rede, mesmo os que não são utilizados. O tráfego gerado na rede é grande, principalmente quando várias estações acessam simultaneamente o servidor.

Já num banco de dados cliente/servidor, a interface com o usuário fica na estação e se comunica remotamente com o mecanismo de acesso, que é um *sistema gerenciador de banco de dados* (SGBD) rodando no servidor. Quando o SGBD recebe um pedido para selecionar alguns dados, ele acessa localmente os dados no servidor e retorna apenas o resultado pedido. No caso de uma atualização, não é necessário nem mesmo retornar um resultado, apenas informar que a atualização foi feita.

O diagrama abaixo resume as diferenças entre os ambientes:

Desktop cliente/servidor



O SQL Server, como já foi dito, é um sistema de gerenciamento de bancos de dados cliente/servidor.

2 - Instalação e Configuração

Requisitos de Hardware e Software

Opções usadas na instalação

Instalando o software de servidor

Verificando a instalação

Instalando o software de cliente

Registrando um servidor

Solução de problemas de instalação

Removendo o SQL Server 7.0

Objetivos:

- Saber o que é necessário para instalar o SQL Server em um computador;
- Aprender a instalar o SQL Server em um servidor Windows NT e a configurar as estações de rede para utilizá-lo.

Requisitos de Sistema

Antes de instalar o SQL Server, é preciso saber quais os requisitos mínimos e recomendados para a instalação.

Computador: Intel e sistemas compatíveis, ou DEC Alpha e compatíveis:

É recomendável que todos os componentes de hardware escolhidos estejam listados na HCL (lista de compatibilidade de hardware) do Windows NT.

Memória: Mínimo de 32 Mb. Recomendável memória adicional, especialmente se o servidor já estiver processando outras funções além de banco de dados, ou se forem usados bancos de dados grandes e replicação

Sistema Operacional: O SQL Server pode ser instalado no Windows NT 4.0 ou superior, com o Service Pack 3 ou posterior, nas plataformas de hardware citadas acima, ou no Windows 9x (no NT ele roda como um serviço e no 9x como uma aplicação). O software de cliente, para acesso ao SQL Server nas estações, pode ser instalado em Windows NT Server, Windows NT Workstation, Windows 95/98, ou Windows 16-bits (3.x), MS-DOS, UNIX, Macintosh, ou navegadores Internet.

Espaço em disco: Numa instalação mínima, são usados 70 MB, e numa instalação completa, 160 MB, incluindo todos arquivos de programas, documentação online, ferramentas de desenvolvimento, e arquivos de exemplo. Uma instalação de um novo servidor, só com as ferramentas de gerenciamento, exige 70 MB

Software de rede: Numa rede Windows NT, o SQL Server usa o software de rede integrado. Não é necessário software adicional, exceto para conectar a alguns outros

tipos de rede. No caso da Novell Netware, o suporte é fornecido pelo protocolo NWLink do Windows NT.

Não é recomendado instalar o SQL Server num computador que seja controlador primário de domínio (PDC) ou controlador reserva (BDC).

Outras considerações

Sistemas de arquivos: o Windows NT suporta FAT ou NTFS. O SQL Server pode usar qualquer um dos dois indiferentemente e o tipo de sistema usado não afeta seu desempenho (exceto quando se usa compressão no NTFS, que reduz o desempenho). Recomenda-se o NTFS para a instalação do servidor, pelas suas vantagens de recuperação e segurança..

Nome do Servidor: o nome do computador usado para o SQL Server deve seguir certas regras:

- Nomes não devem ter mais de 128 caracteres.
- Nomes não podem conter espaços.
- Nomes devem começar com uma letra (maiúscula ou minúscula) ou sublinhado ('_'). Nomes também podem começar com @ ou #, mas como nomes começados por essas letras têm um significado especial, não serão usados na maioria dos casos.
- Caracteres no nome depois da primeira letra, podem ser qualquer letra, número, ou os símbolos @, _, #, ou \$.

Opções Usadas na Instalação

Informação de Registro

Nessa fase, informe seu nome, nome da empresa e número de identificação do produto.

Escolha do tipo da instalação

O SQL Server oferece três tipos de instalação: Típica (Typical), Mínima (Minimal), e Personalizada (Custom).

Instalação típica

Automaticamente instala o SQL Server e utilitários de cliente com as opções padrão de instalação. São instalados: SQL Server, Book Online, Quick Tour, e What's New. Essa opção exige cerca de 163 MB de espaço em disco, e não instala o software de Full Text Search, as ferramentas de desenvolvimento ou os arquivos de exemplo.

Instalação Compacta

Essa opção instala o mínimo de arquivos necessários para operar o SQL Server, e usa todas as opções padrão de instalação. Uma instalação compacta é como uma instalação típica, exceto que as ferramentas de gerenciamento, os livros on-line, Quick Tour e What's New não são instalados. Esta opção exige cerca de 74 MB de espaço em disco.

Instalação Personalizada

Essa opção permite fazer a escolha de quais componentes exatamente serão instalados, além de permitir escolher muitas opções diferentes do SQL Server, ao invés de apenas aceitar automaticamente as opções padrão.

Instalação remota

É possível instalar o SQL Server remotamente em outro computador. Nesse caso, você pode informar o nome do computador e as localizações do Windows NT e da unidade onde o SQL Server será instalado.

Diretórios

O SQL Server é instalado por default no diretório C:\MSSQL, mas isso pode ser alterado. O nome do diretório pode ser longo, mas não deve conter espaços. Dentro desse diretório, todos os arquivos possuem nomes curtos (FAT 8.3), mesmo num drive que suporta nomes longos. Nessa apostila, ao fazermos referência a C:\MSSQL, note que você pode ter instalado em um diretório diferente. Nesse caso, substitua o nome pelo do seu diretório.

O SQL cria os seguintes subdiretórios durante a instalação:

- BACKUP - Contém arquivos de backup.
- BINN - Contém arquivos executáveis das ferramentas de administração do SQL e arquivos do Help Online, e DLLs.
- FTDATA - Só é instalado se o componente de procura por texto completo [Full-Text Search] tiver sido instalado. Utilizada para lidar com arquivos de catálogo somente de texto.
- DEVTOOLS - Essa pasta, e suas subpastas, só são criadas, se escolher-se instalar as ferramentas opcionais de desenvolvimento e exemplos.
- HTML - Armazena arquivos HTML e arquivos relacionados.
- DATA - Contém os arquivos de dados.
- JOBS - Armazena informações sobre tarefas [Jobs] do SQL Server
- INSTALL - Scripts de instalação e arquivos de saídas.
- LOG - Contém arquivos de log de erro.
- REPLDATA - Diretório de trabalho usado para replicação.
- UPGRADE - Arquivos do assistente de atualização de versão. Não é criada se você não instalar o assistente de atualização.

Conjunto de caracteres, ordem de classificação, e comparação Unicode

O *conjunto de caracteres* [character set] usado determina os tipos de caractere que o SQL reconhece nos seus dados e a *ordem de classificação* [sort order] determina como o SQL Server compara dados em forma de caractere e como eles são classificados. Você deve escolher essas opções durante a instalação. A única forma de alterá-las posteriormente é reinstalando o SQL Server.

Como geralmente são usadas aplicações Windows para acesso aos dados, é altamente recomendável usar o conjunto de caracteres ISO 8859-1 (Code Page 1252), também chamado Latin-1 ou "ANSI". Esses caracteres são os mesmos usados no Windows. (Esse é o conjunto de caracteres default durante a instalação). Existe outros

caracteres como: 850(multilingual) que inclui todos os caracteres da Europa, América do Norte e América do Sul, 437(US English) que contém o alfabeto completo do Estados Unidos, 932(Japanese) contém o alfabeto completo do Japão.

A ordem de classificação determina, por exemplo, se o SQL Server considera ou não as letras acentuadas (ç, é, ã) como diferentes dos caracteres não acentuados. Para a língua portuguesa, é recomendável usar a ordem "Dictionary order, case-insensitive, accent-insensitive" [ordem de dicionário, insensível ao caso, insensível a acentos]. Dessa forma, caracteres acentuados e não-acentuados são tratados da mesma forma, como também letras maiúsculas e minúsculas. Por exemplo, ao pesquisar "CAMARA" no banco de dados, "Câmara" será considerado igual. (Mas 'c' e 'ç', ou 'C' e 'Ç' são diferentes). Existem outras opções para ordem de classificação como: Dictionary order, case-insensitive [ordem de dicionário insensível ao caso], neste caso as letras maiúsculas são tratadas da mesma forma que as letras minúsculas, mas os caracteres acentuados são tratados de forma diferente dos caracteres não acentuados (essa é a ordem de classificação default durante a instalação).

O SQL Server tem a capacidade de armazenar caracteres ASCII padrão, e caracteres Unicode. Os caracteres Unicode são capazes de representar mais de 64000 caracteres diferentes., embora os caracteres ASCII sejam capazes de representar apenas 256 caracteres. Depois de escolher o conjunto de caracteres e a ordem de classificação, você deve escolher uma comparação Unicode [Unicode collation], que funciona como uma ordem de classificação para os caracteres Unicode armazenados no SQL Server. Uma comparação Unicode consiste de um local e vários estilos de comparação. Locais, normalmente nomeados de por países ou regiões culturais, ordenam caracteres de acordo com o padrão naquela área. O programa de instalação do SQL Server vai fornecer uma comparação Unicode padrão, com base no conjunto de caracteres e ordem de classificação que você escolheu. É recomendável que não se altere essa seleção, pois caso ela seja alterada, a migração de Unicode para não-Unicode torna-se mais difícil, e dados Unicode e não-Unicode podem ser ordenados de maneiras diferentes.

Protocolos de Rede

Para cada tipo de cliente de rede, o SQL Server possui uma *Net-library*, um driver que suporta comunicação através desse tipo de rede. As opções disponíveis são:

- *Named Pipes (Netbeui)*: Suporta o protocolo Netbeui. É instalado por default. Você não deve removê-lo pois os utilitários do SQL Server depende desta Net-Library.
- *Multi-protocol*: suporta a conexão através de vários protocolos de rede, suportando também segurança integrada e criptografia (caso a aplicação suporte).
- *NWLink IPX/SPX*: comunica-se com clientes Netware. O SQL Server pode se registrar como um serviço numa rede Netware.
- *TCP/IP sockets*: suporta comunicação através de Windows sockets, por exemplo, com um cliente de Internet.
- *Banyan VINES, AppleTalk ADSP, DECnet*: outros tipos de rede.

Opções de auto-inicialização

Serviço é um programa executável que não tem interface com o usuário, mas tem formas de controlar. Como iniciar e parar o serviço determinado.

Definem se os serviços MSSQLServer e SQLServerAgent iniciarão automaticamente com o Windows NT ou serão iniciados manualmente. Essa opção pode ser alterada depois no Painel de Controle [Control Panel] do Windows NT, opção Serviços [Services].

Para o serviço SQLServerAgent, você pode também definir qual o nome de usuário e senha que ele utiliza para se conectar ao sistema.

Modo de licenciamento

O SQL Server pode ser licenciado *por servidor* [per server], onde para cada servidor adquire-se N licenças de acesso, ou *por estação* [per seat], onde existe uma licença para cada estação, independentemente do número de servidores usados. Durante a instalação você pode definir qual dos modos utilizar. Para alterar essas opções posteriormente, use o ícone Licenciamento [Licensing] no Painel de Controle do Windows NT.

Instalando o software de servidor

Criando uma conta para o SQLServerAgent

Durante a instalação, SETUP pede um nome de conta de usuário e a senha dessa conta, para uso do serviço SQLExecutive. Existem duas maneiras de selecionar o usuário :

- Criar uma conta no Windows NT .
- Utilizar a conta Local System do Windows NT. Mas nesse caso nem todos os recursos do SQLExecutive podem ser usados. Por exemplo:

Ao utilizar Backups poderá não se ter acesso ao driver de destino do Banco de Dados. É recomendável criar uma nova conta de usuário antes de instalar. Essa conta não deve ser usada por usuários para logar no computador. Ela será exclusiva para o SQL Executive.

Para isso, abra o Gerenciador de Usuários [User Manager] do Windows NT, no grupo/menu de programas Ferramentas Administrativas [Administrative Tools]. Clique em **Usuário|Novo usuário...** [User|New user...]. Informe o nome de usuário "SQLExecutive" (o nome pode ser qualquer). Informe uma senha e anote-a para usar mais tarde com o SETUP. Desmarque a opção "O usuário deve alterar a senha no próximo logon" [The user must change password...] e marque "A senha nunca expira" [Password never expires].

Depois acrescente esse usuário ao grupo local "Administradores" [Administrators]. Use o menu **Diretivas|Direitos de usuário...** [Policies|User rights] para conceder os direitos de "Logon como serviço" [Logon as a service], "Agir como parte do sistema operacional" [Act as part of the operating system], "Aumentar cotas" [Increase quotas], "Substituir um token de nível de processo" [Replace a process level token] à conta

'SQLExecutive' . Como são direitos avançados, marque a opção Exibir direitos avançados do usuário [Advanced User Rights].

Iniciando o SETUP

Para instalar o SQL Server 7, efetue login no Windows NT com uma conta que tenha privilégios administrativos, coloque o CD de instalação na unidade de CD . Em alguns segundos, o programa de Instalação do SQL Server irá iniciar automaticamente e mostrar a tela abaixo:



Nessa tela você tem diversas opções. Para iniciar a instalação do SQL Server, selecione "Install SQL Server 7.0 Components". A próxima tela permite que se selecione quais componentes se quer instalar.

Nota: Se você tiver a opção de autoReprodução do CD desativada, você pode iniciar manualmente o programa de instalação do SQL Server. Para fazê-lo, vá para o diretório raiz do CD de instalação do SQL Server, e execute o programa "setup.bat". Então será mostrada a tela acima. Se o SQL Server estiver sendo instalado de um compartilhamento da rede, mude para a pasta do compartilhamento que contem o SQL Server e execute o arquivo "setup.bat".

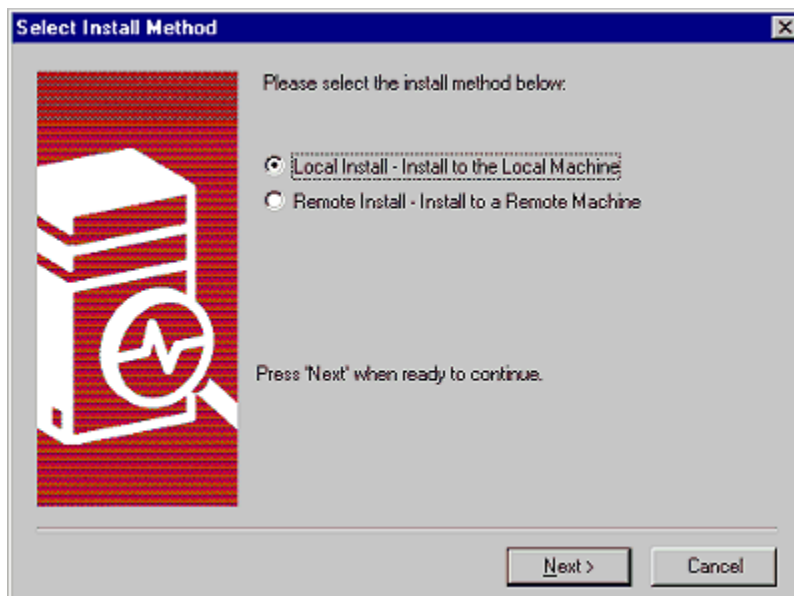
Nota: A opção "Install SQL Server Prerequisites" permite intalar os softwares necessários para se poder instalar o SQL Server. Estes são o service pack 4 do Windows NT e o Internet Explorer 4.01.

Install SQL Server 7.0 Components



Nessa tela, selecione "Database Server- Standard Edition" se você quiser instalar o SQL Server no Windows NT, ou "Database Server- Desktop Edition" se você quiser instalar o SQL Server no windows 9x.

Select Install Method



Se você quisesse instalar em um computador remoto, clicaria em "Remote Install" e informaria as opções do computador remoto. No nosso caso, clique em Next para fazer a instalação local.

Aí, aparece a tela de boas-vindas:



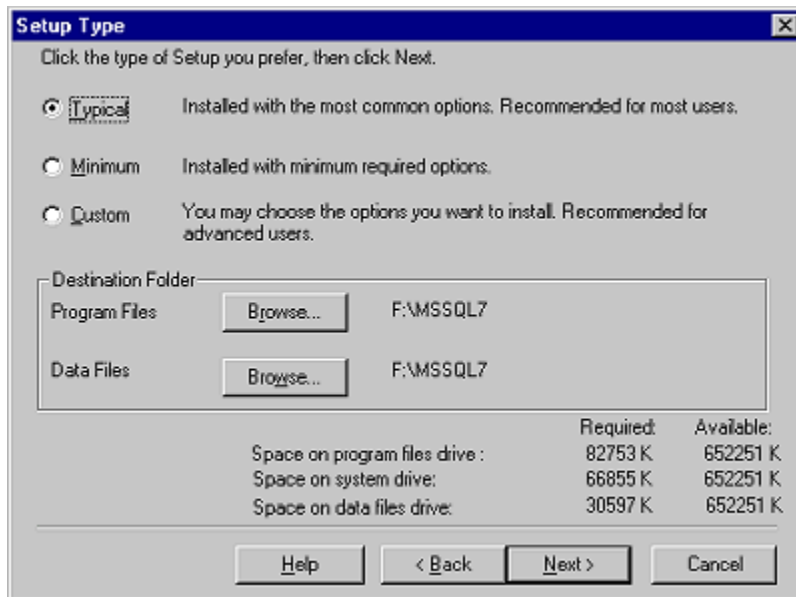
Clique em Next para continuar.

Aparece a tela de "Acordo de Licença de Software" [Software License Agreement]. Se você aceita os termos do acordo, clique em Yes para continuar. Você deve selecionar Yes se você quer instalar o SQL Server.

Aí aparece a tela de informação do usuário. Aqui, entre seu nome e o nome da companhia. Depois que tiver entrado com essas informações, clique em Next para continuar.

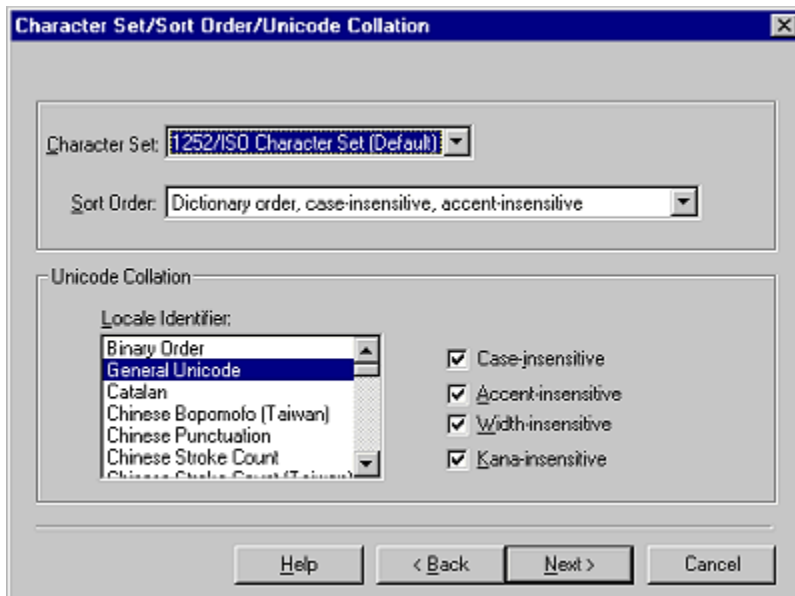
A seguir será pedido o número de série do SQL Server. Este pode ser encontrado no adesivo amarelo colado na caixa do CD. Depois de entrar com esse número, clique em Next para continuar.

Setup Type

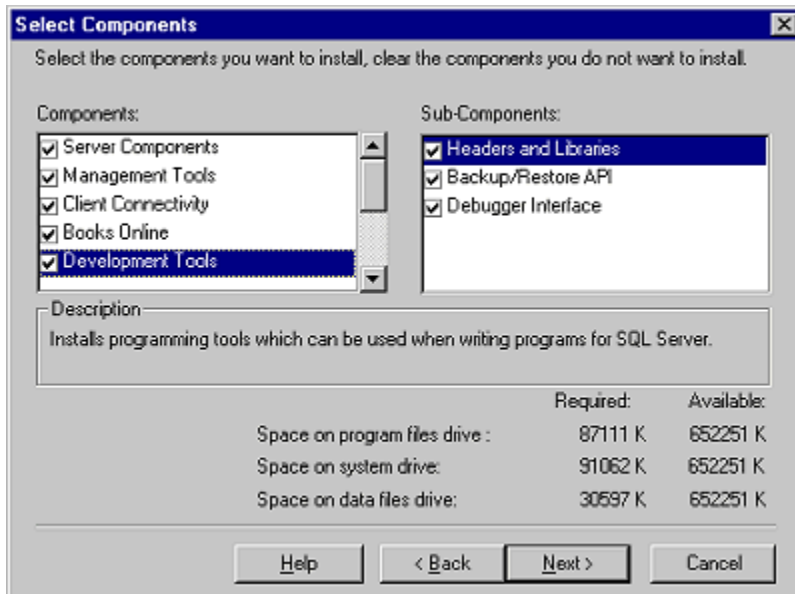


Aqui, você deve selecionar se você quer fazer uma instalação mínima, típica, ou personalizada. Já discutimos sobre cada uma das opções anteriormente. Aqui, usaremos a instalação personalizada, já que ela fornece a maior flexibilidade quando instalamos o SQL Server. Clique no botão perto de "Custom" para escolher a instalação personalizada.

Ainda nesta janela, você deve decidir onde armazenar os arquivos de programa e de dados do SQL Server. Podem ser instalados no mesmo local, ou em locais diferentes. Use os botões "Browse" para selecionar outros locais que não sejam os locais padrão selecionados. Use as informações de espaço exigido [Required] e disponível [Available] para decidir melhor onde instalar os arquivos.



Feitas as seleções, clique em Next para continuar.



Select Components

Aqui, você deve escolher aqueles componentes do SQL Server que você quer carregar. Perceba que no lado esquerdo da tela estão os componentes, e no lado direito os sub-componentes. Primeiro, você seleciona um componente clicando dentro da caixa de verificação, e se houver subcomponentes, você os escolhe clicando em caixas de verificação do lado direito. Enquanto você escolhe os componentes para sua instalação, perceba que você pode determinar o tamanho da instalação observando o

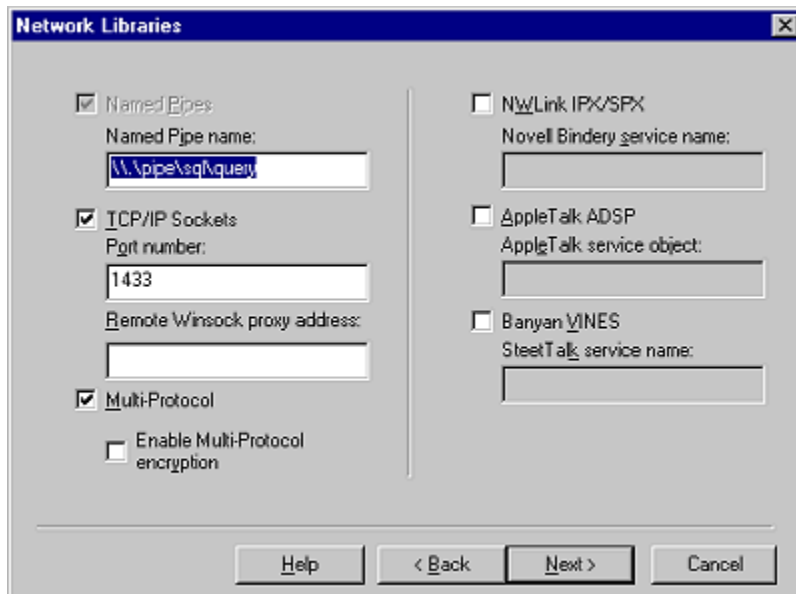
espaço disponível em disco na parte inferior da janela. Depois de escolhido o que for apropriado, clique em Next para continuar.

Character Set/ Sort Order/ Unicode Collation

Aqui você escolhe tanto o conjunto de caracteres [Character Set] quanto o ordem de classificação [Sort Order] da caixa de lista correspondente no topo da tela. Conforme já discutido, recomenda-se deixar o padrão para o "Character Set" e pôr a "Sort Order" como "Dictionary order, case-insensitive, accent-insensitive".

Como já citado, a não ser que você tenha uma razão realmente boa para mudar essa opção, deixe a opção padrão selecionada. Depois de terminar as suas escolhas, clique em Next para continuar.

Considerações importantes: Um conjunto de caracteres pode ser modificado depois da instalação, mas exige que você reconstrua todos seus bancos de dados e recarregue os dados (uma tarefa dispendiosa!). Também é recomendável que todas as instalações do SQL Server que precisem comunicar-se usem o mesmo conjunto de caracteres, e ordem de classificação, ou você poderá ter resultados inesperados.

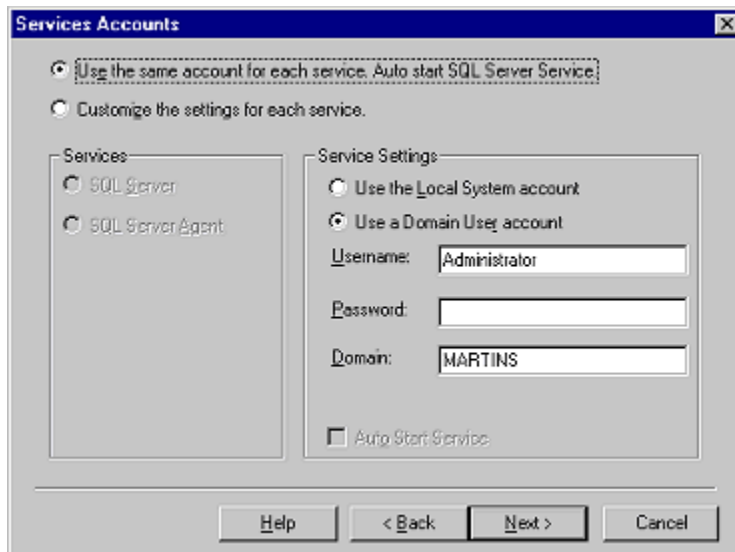


Network Libraries

Nessa janela, de seleção das bibliotecas de rede [Network Libraries], você deve, para cada biblioteca de rede que for instalar, entrar com alguns parâmetros adicionais, como a porta em que o servidor vai escutar, para a biblioteca de rede TCP/IP, e outras de acordo com a biblioteca de rede a ser instalada. Como já foi dito, a biblioteca de rede Named Pipes deve ser selecionada, pois é utilizada durante a instalação. Depois da instalação concluída, ela pode ser retirada, apesar de não ser recomendável. Normalmente, as opções padrão funcionam bem. Depois de feitas as seleções necessárias, clique em Next para continuar.

Nota: Para mudar uma net-Library depois da instalação, use o "SQL Server Network Utility" (SRVNETCN.EXE)

Services Accounts



Aqui você pode usar a mesma conta para os dois serviços (SQLServer e SQLServerAgent), iniciando automaticamente o serviço SQL Server. Para isso, selecione o primeiro botão [Use the same account for....]. Caso você queira usar uma conta diferente para cada serviço, selecione "Customize the settings for each service" e os botões SQL Server e SQL Server Agent se tornarão ativos. Aí as opções do serviço [Service Settings] serão referentes ao serviço selecionado.

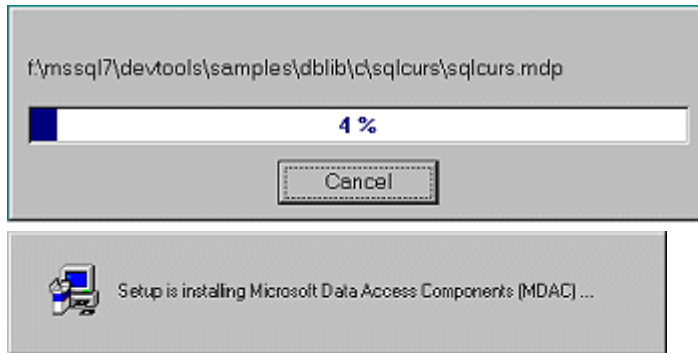
Se foi criada a conta do usuário para uso do SQLServerAgent, informe o nome do usuário, a senha e confirme a senha (o default é o nome de usuário conectado atualmente). Informe também o domínio do NT Server onde essa conta existe. Caso não tenha criado a conta marque a opção Install to log on as Local System account (é uma conta predefinida do Windows NT).

Nota: Caso se escolha usar a conta de sistema local, o SQL Server não será capaz de se comunicar com outros servidores.

Clique em Next para continuar.

A seguir aparece a tela de seleção do modo de licenciamento [Choose Licensing Mode]. Escolha o licenciamento Per Server ou Per Seat, levando em conta o que foi discutido anteriormente. Clique em Next para continuar.

"Copying Files" e "SETUP is..."



...

O SETUP vai copiar os arquivos necessários para o diretório de instalação, reindexar as tabelas de sistema, e definir a configuração inicial do SQL Server. Esse processo leva cerca de 10-15 minutos, dependendo da velocidade do seu computador. Após esse processo, o SQL Server terá sido instalado e estará pronto para usar. As ferramentas do SQL Server estarão disponíveis no submenu "Microsoft SQL Server 7.0", dentro do menu Iniciar|Programas [Start|Programs] do Windows NT 4.0. Você precisará de iniciar seus serviços antes de poder conectar-se pela primeira vez ao servidor SQL Server.

Depois que o SQL Server estiver instalado, há alguns passos a serem completados antes de se poder dizer que o SQL Server está pronto para rodar. Alguns desses passos devem ser executados apenas uma vez para se assegurar que o SQL Server foi instalado corretamente e para prepará-lo para rodar corretamente no futuro. Aqui vamos citar esses passos. Outros passos, que incluem configurações, ajustes e otimização, estabelecimento de IDs para login e IDs de usuários do banco de dados, e é claro, a criação de novos bancos de dados, serão discutidos em outras seções. Nesta seção o objetivo será verificar se o SQL Server foi instalado corretamente.

Verificando se o SQL Server foi instalado corretamente

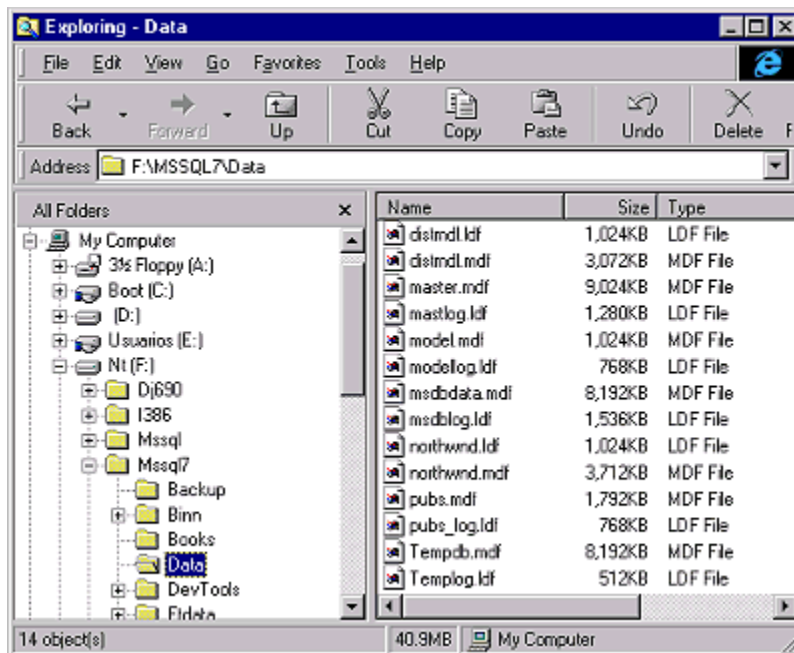
Depois que o SQL Server estiver instalado, você deve gastar alguns minutos para verificar que o mesmo foi instalado e está rodando adequadamente. Use os passos abaixo para verificar a instalação.

Verifique se as pastas de programas e dados foram criadas

Use o NT Explorer (ou Windows Explorer) para verificar que os arquivos de programas do SQL Server e os arquivos de dados foram instalados nas pastas que você especificou durante o processo de instalação.

Cada pasta deve conter pastas adicionais de acordo com o que foi dito anteriormente (em Diretórios). Verifique se todas as subpastas tanto nas pastas de programas quanto de arquivos existem.

Deve-se verificar o conteúdo da subpasta /data, que está abaixo da pasta de dados que você especificou. É aí que o SQL Server cria e armazena vários bancos de dados padrão e logs de transações. A pasta /data deve se parecer com a figura abaixo.



Os arquivos são:

- Distmdl.mdf e Distmdl.ldf (apenas se objetos opcionais de replicação forem instalados).
- Northwnd.mdf e Northwnd.ldf
- Master.mdf e Mastlog.ldf
- Model.mdf e Modellog.ldf
- Msdbdata.mdf e Msdbbolog.ldf
- Pubs.mdf e Pubs_log.ldf
- Tempdb.ldf e Templog.ldf

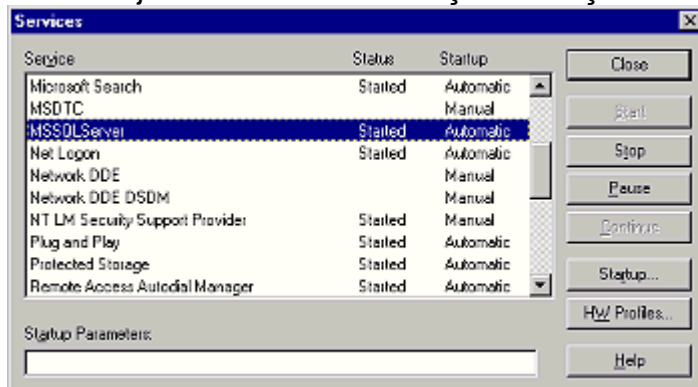
Verificar que todas as ferramentas de gerenciamento do SQL Server estão instaladas

Verifique se as ferramentas de gerenciamento do SQL Server que você especificou durante a instalação foram instaladas. O modo mais fácil de fazer isso é clicando em Iniciar | Programas | Microsoft SQL Sever 7.0. Aí você verá um menu com a listagem de todos os programas, utilitários e documentação online que você especificou durante a instalação.

Verifique que os serviços do SQL Server estão carregados e executando

Quando você instalou o SQL Server, seus dois serviços principais - SQLServerAgent e MSSQLServer - foram instalados e configurados para iniciar automaticamente (assumindo que você quis inicialização automática). Quando o SQL Server é instalado pela primeira vez, os dois serviços não são iniciados automaticamente até que o NT Server seja reiniciado.

Nota: Veja mais sobre os serviços na seção do Service Manager.



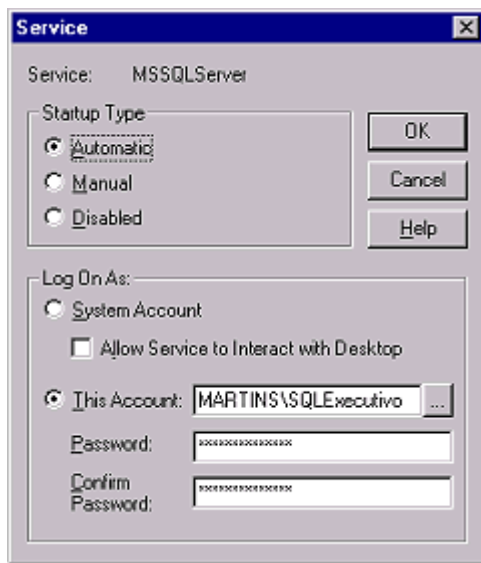
Conferindo se os dois serviços foram instalados, e manualmente iniciando-os pela primeira vez

1. Abra o Painel de Controle do NT, e então dê um duplo clique no ícone Serviços. Aparece então a caixa de diálogo de Serviços.
2. Percorra a lista de serviços até que você veja pela primeira vez o serviço MSSQLServer. Se você não reinicializou o computador desde que instalou o SQL Server, ele deve ter um Estado [Status] em branco. E em Inicialização [Startup], deve estar configurado como automático.
3. Para iniciar manualmente o serviço, clique nele uma vez para que o mesmo fique selecionado, e então clique no botão Iniciar [Start]. Depois de uns 30 segundos, o Estado [Status] vai mudar para Iniciado [Started] e o serviço está agora sendo executado.
4. Percorra a lista um pouco mais até que você veja o serviço SQLServerAgent, e ele também deve estar com o Estado em branco e a Inicialização como Automática.
5. Para iniciá-lo manualmente, faça da mesma maneira que foi feito com o serviço MSSQLServer, mas agora selecionando o serviço SQLServerAgent. Depois de

uns 15 segundos, o Estado [Status] vai mudar para Iniciado [Started] e o serviço está agora sendo executado.

6. Se tudo estiver correto, feche a caixa de diálogo de Serviços e o Painel de Controle.

E se os serviços não estiverem aí? Se os serviços não estiverem listados, sua instalação do SQL Server falhou e você vai ter que reinstalá-lo, mas isso é raro. Você pode ter notado que na coluna Inicialização [Startup], estava a a palavra Manual, ao invés de Automático. Se você se vir nessa situação, você provavelmente esqueceu de escolher a opção auto-iniciar [Auto-start] quando você instalou o SQL Server. É bem fácil corrigir isso.



Escolhendo auto-iniciar

1. Um de cada vez, selecione cada um dos serviços do SQL Server na caixa de diálogo de Serviços e clique no botão Inicialização [Startup]. Isso mostra a janela abaixo.
2. Nessa janela, selecione Automático como o tipo de Inicialização [Startup type].
3. Clique em OK, e você retornará para a caixa de diálogo de Serviços.
4. Repita os passos 1 a 3 para o outro serviço do SQL Server.
5. O último passo é iniciar manualmente os serviços como descrito anteriormente. Se os serviços não iniciarem conforme descrito, e se você receber uma mensagem de erro, veja a solução de problemas de instalação para tentar resolver o problema.

Verifique que você consegue se logar no SQL Server

A última maneira de verificar que o SQL Server foi instalado corretamente é tentar se logar e executar uma pequena consulta. Se você puder executar essas duas tarefas com sucesso, você saberá que o SQL Server foi instalado sem problemas.

Aqui vamos demonstrar como se logar ao servidor usando a ferramenta ISQL, que é o programa baseado em linha de comando fornecido com o SQL Server, utilizado para

executar comandos Transact-SQL. Embora haja outros programas ou utilitários do SQL Server que você poderia usar ao invés do ISQL, ele é recomendado para este teste porque há menos coisas para dar errado quando da execução do teste.

Execute e se logue ao ISQL, e rode uma pequena consulta para verificar que o SQL Server foi instalado corretamente.

1. Vá para o prompt de comando do NT.
2. No prompt de comando, escreva o seguinte e pressione Enter:
`isql /Usa /P`
3. Se tudo estiver funcionando corretamente, o prompt do ISQL deverá aparecer ao invés do prompt de comando. Ele se parece com isso:
`1>`
4. Escreva a seguinte consulta para testar se o SQL Server vai responder. Pressione Enter depois de cada linha.
`select @@servername`
`go`
5. Assumindo que tudo esteja funcionando, o nome do seu servidor deve ser mostrado no prompt do ISQL. A resposta deve se parecer (o número e nome do servidor vai variar) com isso:
`PDC`
`(1 row affected)`
`1>`
6. Para sair do programa ISQL, digite **exit** no prompt do ISQL e pressione Enter, e você retornará para o prompt de comando do NT.

Se tudo tiver ocorrido como descrito, você sabe que o SQL Server foi instalado corretamente e está funcionando adequadamente. Se você encontrar problemas ou mensagens de erro, veja a solução de problemas de instalação para tentar resolver o problema.

Instalando o software de cliente

Quando você já tiver instalado o SQL Server, é hora de pensar em instalar as ferramentas de gerenciamento nas estações de trabalho que irão ser utilizadas para administrar remotamente o SQL Server. Também pode ser interessante instalar as ferramentas de gerenciamento nas estações de trabalho dos desenvolvedores.

As ferramentas de gerenciamento são as mesmas que você teve a chance de instalar quando da instalação do SQL Server (afinal, mesmo na máquina onde o servidor está instalado, você precisa de software de cliente para conectar ao servidor). A vantagem de instalar as ferramentas de gerenciamento em outras máquinas é que você pode gerenciar remotamente o SQL Server de virtualmente todos computadores, não apenas do servidor físico em que o SQL Server está sendo executado.

As ferramentas de gerenciamento do SQL Server podem ser instaladas no NT Server, NT Workstation, e Windows 9x. Ao contrário de algumas das ferramentas de gerenciamento incluídas no SQL Server 6.5, as ferramentas de gerenciamento do SQL Server 7.0 não podem ser executadas sob o Windows 3.x ou DOS.

Aqui descreveremos as ferramentas de gerenciamento, como instalar e utilizá-las.

Todas as ferramentas de cliente aqui descritas são as mesmas descritas em

Ferramentas de gerenciamento, mas algumas delas, especificamente o Service Manager e o Server Network Utility só são instaladas no servidor. Onde se vir escrito "Ferramentas de gerenciamento" nesta seção, subentende-se que se está citando as ferramentas de cliente (que nada mais são do que as ferramentas de gerenciamento instaladas em um cliente).

Nota: Qualquer das ferramentas de gerenciamento só podem ser utilizadas por um usuário que tenha as permissões necessárias para tal tarefa.

Ferramentas de cliente

O SQL Server inclui uma porção de ferramentas de administração para serem instaladas no cliente, que podem ser usadas para gerenciar o SQL Server. Durante o processo de instalação, você pode instalar quantas ferramentas de gerenciamento você achar necessário. O SQL Server oferece estas opções:

- **SQL Server Enterprise Manager:** Se você quer administrar remotamente o SQL Server, então a instalação do Enterprise Manager é necessária. Este programa permite ao DBA executar virtualmente qualquer tarefa administrativa no SQL Server.
- **SQL Server Profiler:** Esta ferramenta é usada para monitorar e registrar a atividade dos bancos de dados entre o SQL Server e os clientes. Apenas instale esta ferramenta nas máquinas que executarão esta tarefa.
- **SQL Server Query Analyzer:** Esta ferramenta é utilizada para enviar manualmente comandos Transact-SQL e procedimentos armazenados para o mecanismo de banco de dados do SQL Server. Você provavelmente instalará esta ferramenta na maioria, talvez em todas as máquinas de gerenciamento remoto.
- **Client Diagnostic Utilities:** Esta ferramenta é utilizada para verificar qual DB-Library está instalada em um cliente, e para configurar as ferramentas de gerenciamento para se comunicar com o SQL Server em uma rede.
- **MS DTC Client Support:** A ferramenta de Coordenação de Transações Distribuídas (DTC) da Microsoft fornece suporte aos clientes DTC. Apenas necessita ser instalada em clientes que executem uma aplicação do SQL Server que exijam o DTC.
- **Development files:** Esses arquivos são necessários para desenvolvedores OLE-DB para a criação de programas utilizando ODBC, DB-Library, ODS, SQL-DMO, Embedded SQL for C, e MS DTC. Geralmente, apenas desenvolvedores SQL Server precisarão desses arquivos.
- **Sample files:** Estes arquivos são arquivos de exemplo feitos para os desenvolvedores analisarem e aprenderem com eles. Assim como os arquivos de desenvolvimento, apenas desenvolvedores precisarão deles.
- **Replication Conflict Resolution Tool:** Utilizada para ajudar a resolver conflitos de replicação entre dois servidores SQL Server. Você apenas precisa desta ferramenta se você implementar replicação em seus servidores.
- **Livros online:** Esta é uma documentação completa e abrangente do SQL Server, e é interessante instalá-la em todo cliente. Ocupa cerca de 15 MB de

espaço no disco, mas vale a pena. Se você não quiser ocupar todo esse espaço em um cliente, você também pode instalar os livros online em um compartilhamento de rede, e então conectar a ele quando necessário. Você também tem a opção de executá-lo a partir de um drive de CD instalado localmente.

Não importa quais das opções acima você decidir instalar, as seguintes ferramentas e arquivos são instalados automaticamente. eles incluem bcp, isql, osql, ODBC, e DB-Library. Falaremos deles mais tarde.

Software e Hardware necessário para a instalação do software de cliente

- **CPU:** No mínimo um Alpha AXP ou um Intel de 32 bits (80486). Recomenda-se um Pentium 200 ou mais veloz.
- **RAM:** No mínimo 32 MB. Recomenda-se 32 MB no Windows 9x, e 64 MB ou mais no Windows NT.
- **Monitor e placa de vídeo:** Qualquer placa de vídeo que esteja instalada corretamente funcionará. Recomenda-se pelo menos um monitor de 15" com resolução de 1024x768.
- **Espaço em disco:** 73 MB para a instalação de todas as ferramentas de gerenciamento. Como provavelmente não serão instaladas todas elas, pode ser suficiente menos espaço.
- **Drive de CD-ROM:** Apenas exigido se a instalação estiver sendo feita a partir de um CD.
- **Placa de rede:** Qualquer placa de rede que funcione no sistema operacional será aceita. Recomenda-se uma placa de 10 ou 100 Mbits se você estiver acessando um ou mais servidores SQL Server pesadamente a partir da estação trabalho.
- **Sistema operacional:** No mínimo NT Server ou Workstation 4.0 (com Service Pack 3), ou Windows 9x. Recomenda-se usar sempre o Service Pack mais recente em qualquer desses sistemas.

De onde instalar as ferramentas de gerenciamento

As ferramentas de gerenciamento do SQL Server podem ser instaladas a partir dos seguintes locais.

- **CD de instalação do SQL Server:** As ferramentas de gerenciamento podem ser instaladas diretamente do CD executando-se o programa de instalação do SQL Server. Este é o mesmo programa utilizado para instalar o SQL Server.
- **Compartilhamento de rede:** O meio mais flexível de se instalar as ferramentas de gerenciamento é instalá-las a partir de um compartilhamento na sua rede. O processo de instalação se inicia quando se executa o arquivo setup.bat, o mesmo utilizado para instalar o SQL Server. Se você pretende instalar diversas cópias das ferramentas de administração em várias máquinas, este é o meio mais eficiente.

Como instalar as ferramentas de gerenciamento

A instalação das ferramentas de gerenciamento do SQL Server é um processo simples. Na verdade, os passos exigidos para a instalação das ferramentas de gerenciamento são praticamente os mesmos necessários para instalar o SQL Server. Isso significa que você já conhece o processo. Abaixo listam-se os passos necessários para a instalação das ferramentas de gerenciamento do SQL Server em um cliente.

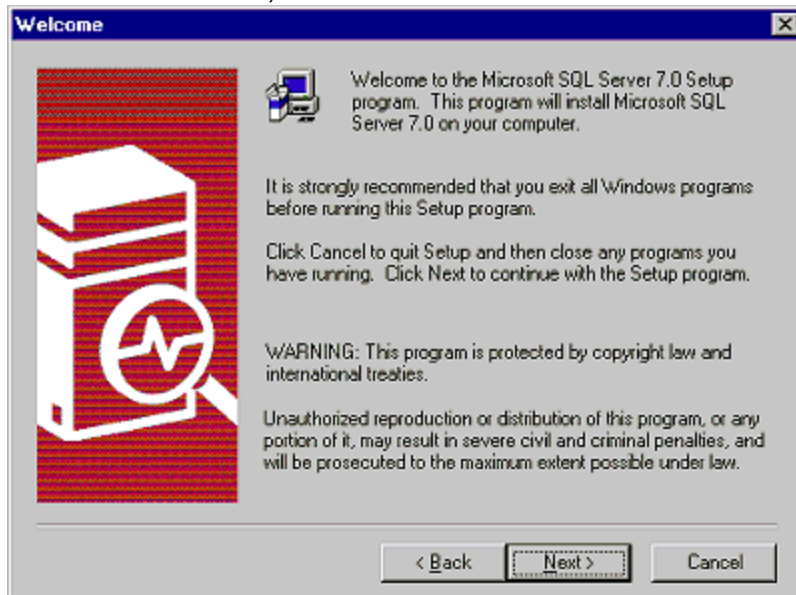
1. Se você estiver instalando as ferramentas de cliente no Windows NT, você deve efetuar logon com uma conta que tenha privilégios administrativos. Se você estiver instalando-as no Windows 9x, você pode efetuar logon com qualquer conta de usuário.
2. Certifique-se de que não há nenhum outro programa sendo executado, antes de iniciar o processo de instalação. Se houver, feche-os antes de continuar. Confirme também se há alguma versão antiga das ferramentas de gerenciamento instaladas na máquina. Se houver, remova-as antes de instalar a nova versão.



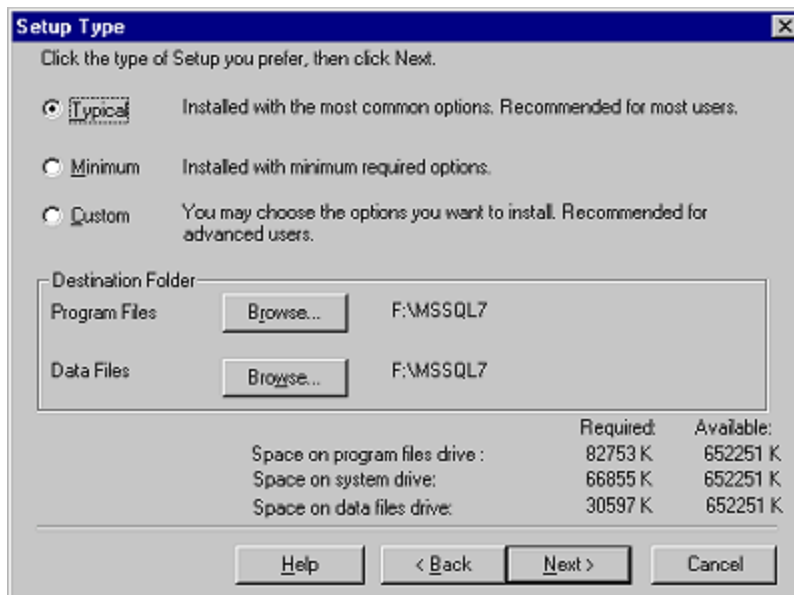
3. Se você estiver fazendo a instalação a partir de um CD, apenas insira o CD do SQL Server no drive de CD. Em alguns segundos, a tela do programa de instalação do SQL Server vai surgir automaticamente. Caso você não esteja instalando de um CD, execute o arquivo **setup.bat** do compartilhamento de rede onde estão os arquivos de instalação. Aparece a tela abaixo.

Nesta tela, você tem uma porção de opções. Selecione Install SQL Server 7.0 Components.

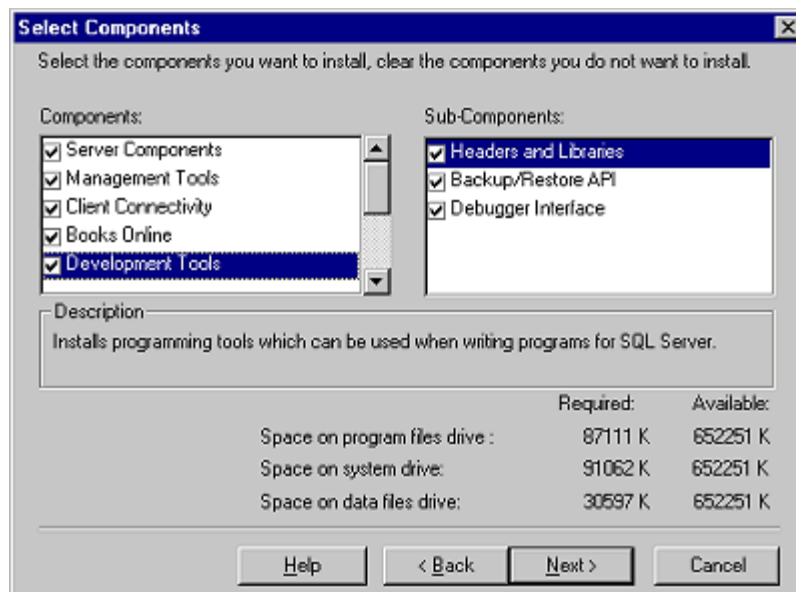
4. Na próxima tela, escolha Database Server- Desktop Edition.
5. Depois você deve escolher se vai fazer uma instalação para a máquina local ou para uma máquina remota. Supondo que você esteja instalando as ferramentas de gerenciamento estando fisicamente na máquina em que elas serão executadas, escolha Local Install.



6. Depois de passar por estas telas introdutórias, aparece a primeira tela da instalação do SQL Server.
7. O programa de instalação é um assistente que te encaminha pelo processo de instalação. Embora o processo seja o mesmo que o da instalação do SQL Server, você não repetirá exatamente os mesmos passos. Aqui, você quer instalar apenas as ferramentas de gerenciamento. Clique em Next para continuar.
8. Aparece a tela do acordo de licença de software. Clique em Yes para continuar.
9. Agora, você deve digitar seu nome e o nome da sua organização, e a seguir o número de série do produto. Clique em Next para continuar.
10. Agora, aparece a tela de tipo da instalação [Setup Type].



11. Para instalar apenas as ferramentas administrativas, escolha Custom e clique em Next. Antes de prosseguir, verifique se as pastas escolhidas estão corretas, e se você tem espaço em disco suficiente. Se não, mude as pastas ou discos, clicando no botão Browse.



12. Na tela de seleção de componentes, você deve selecionar os componentes que você deseja instalar e desmarcar os que você não instalará.

Geralmente, você desmarcará Server Components (afinal, você só quer as ferramentas de administração, e não o SQL Server). Selecione Client Connectivity, e Management Tools. Do lado direito da tela, você pode desmarcar componentes individuais das mesmas. Depois que você escolher o que achar adequado, clique em Next para continuar.

13. Finalmente, a instalação vai começar. Lhe é mostrada a tela final, onde você pode conferir as opções que você selecionou. Se estiver satisfeito com as seleções feitas, clique em Next para começar a instalação. Caso contrário, clique em Back, e altere o que achar necessário.

Depois que você clica em Next, os arquivos são copiados para seu computador.

Quando o processo de instalação finalizar, você volta para a primeira tela do programa de instalação. Para sair dessa tela, clique em Exit.

Quando as ferramentas de gerenciamento já estiverem instaladas, você pode testar se a instalação foi bem-sucedida. Veja, se em Iniciar | Programas | Microsoft SQL Server 7.0, as ferramentas que você selecionou tem seus atalhos. Para verificar se as ferramentas funcionam, escolha alguma e a execute, vendo se você pode se conectar a um servidor remoto. Claro, para que este teste funcione, você deve ter uma conta legal e as permissões necessárias no servidor SQL Server. Se você puder fazer uma conexão com o servidor SQL Server, você confirma que as ferramentas foram instaladas com sucesso.

Na maioria dos casos, assim que as ferramentas de gerenciamento forem instaladas, você será capaz de usá-las imediatamente sem qualquer configuração adicional. Mas se a sua rede não for uma rede Microsoft pura, então você pode ter que mudar algumas opções de configuração usando a ferramenta de configuração de clientes [Client Network Utility]. Veja na seção Client Network Utility como utilizar essa ferramenta.

Registrando um servidor

Quando você tiver certeza que o SQL Server está instalado e você pode se conectar a ele (confira em verificando se você pode se logar no SQL Server), a próxima etapa é se registrar com o SQL Enterprise Manager. Enterprise Manager é o principal programa usado para gerenciar o SQL Server (veja mais sobre o SQL Enterprise Manager). Ele tem a capacidade de não apenas gerenciar um servidor SQL Server local, mas também servidores múltiplos conectados em qualquer lugar na mesma rede física. Mas antes de tornar o SQL Enterprise Manager capaz de fazer isso, você deve lhe contar sobre os diversos servidores SQL Server que você possa ter, e isso é feito através do processo de registrar-se. Quando um servidor SQL Server está registrado com uma cópia do Enterprise Manager, essa cópia do SQL Enterprise Manager é capaz de gerenciar aquele servidor SQL Server, não importando onde ele esteja na rede.

Nota: Por padrão, para administrar um servidor, você deve ser membro do grupo local Administradores no computador onde o SQL Server foi instalado. Administradores de domínio do Windows NT são membros do grupo local Administradores.

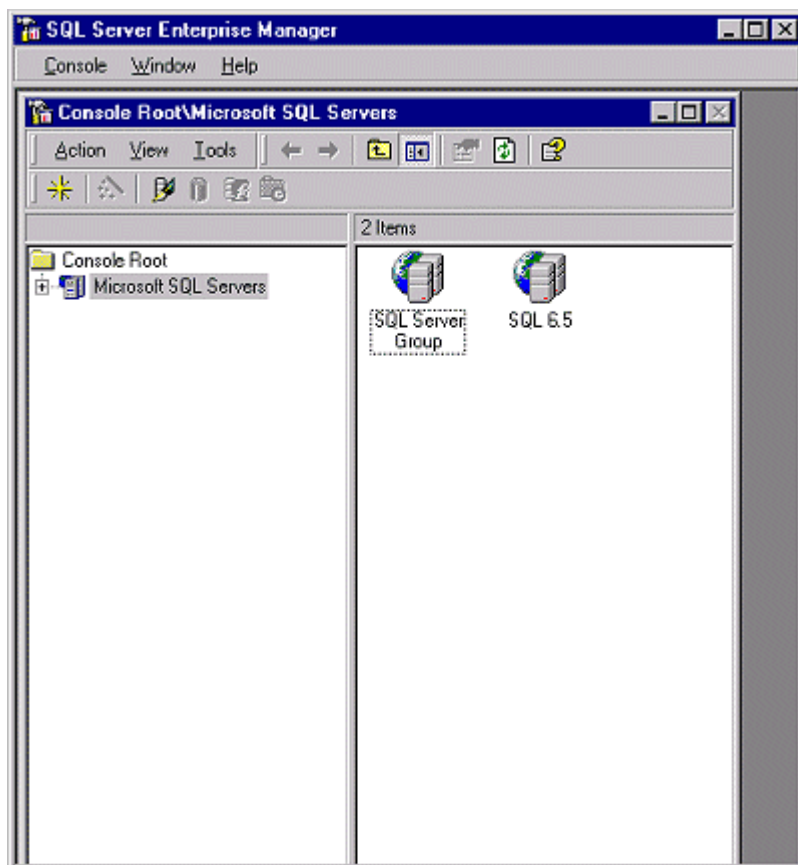
O registro é um processo que precisa ser feito apenas uma vez, embora você possa desregistrar e registrar novamente qual servidor SQL Server quantas vezes você quiser. As informações de registro do SQL Server são mantidas no registro do Windows NT. O Enterprise Manager usa essa informação a cada vez que você se conecta com um servidor SQL Server registrado.

Há duas maneiras de registrar um servidor SQL Server usando o Enterprise Manager: manualmente, ou com o Assistente de Registro [Registration Wizard]. Aqui olharemos os dois métodos, começando pelo Assistente de Registro.

Nota: Diversas cópias do Enterprise Manager podem estar distribuídas em diversos computadores ao longo da empresa. Isso pode fazer com que seja necessário você registrar diversas vezes o(s) servidor(es) que você quer administrar. Isso ocorre pois o registro é feito com o Enterprise Manager e portanto, em cada cópia do Enterprise Manager devem ser registrados os servidores que se deseja administrar.

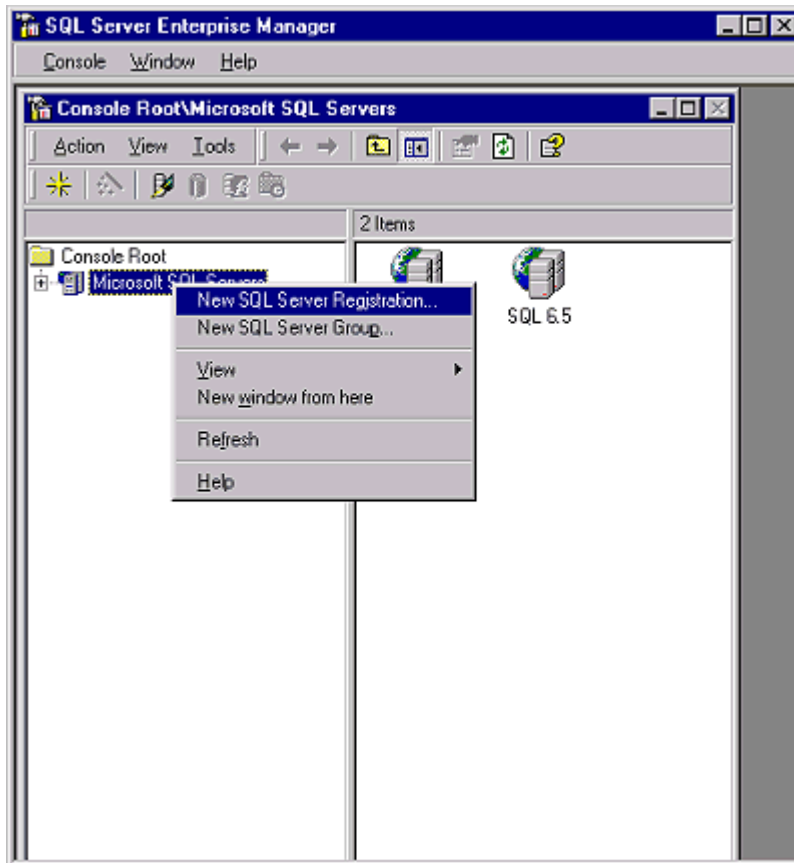
Registrando um servidor utilizando o Registration Wizard

Antes de tentar registrar um servidor SQL Server com o SQL Enterprise Manager, confirme que o SQL Server está instalado e rodando adequadamente (ver em verificando a instalação). Se ele não estiver instalado e rodando corretamente, você não será capaz de registrá-lo com o SQL Enterprise Manager.



Registrando um servidor SQL Server com o SQL Enterprise Manager pela primeira vez.

1. A partir do Grupo de programas do Microsoft SQL Server (Iniciar | Programas | Microsoft SQL Server 7.0 | Enterprise Manager), execute o SQL Server Enterprise Manager. Isso inicia o Microsoft Management Console (MMC).
2. Agora você está pronto para iniciar o assistnte de registro [Registration Wizard]. Para iniciá-lo, clique com o botão direito em Microsoft SQL Servers, que aparece



abaixo de Console Root. Aparecem então diversas opções, e você quer registrar um servidor [New SQL Server Registration...]



3. Depois de escolher a opção New SQL Server Registration, aparece a primeira tela do Assistente de Registro [Registration Wizard]. Clique em

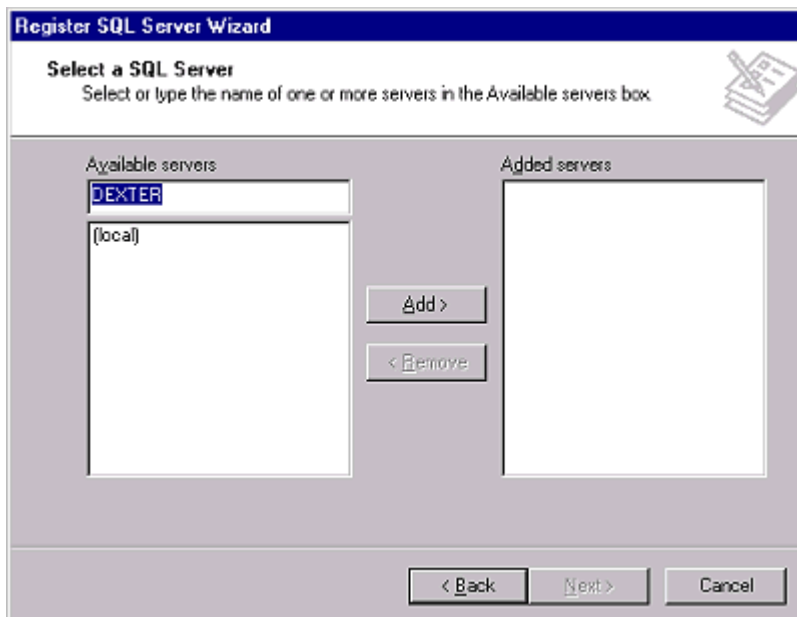
Next para continuar.

4. A seguir, o assistente te pede para selecionar qual servidor SQL Server que você quer registrar, como na figura abaixo:

Os servidores SQL Server disponíveis aparecem no lado esquerdo da tela. Se o seu servidor não estiver sendo mostrado, talvez ele não tenha sido instalado adequadamente, ou os dois serviços do SQL Server podem não estar iniciados (ver em verificando a instalação se eles estão iniciados). Supondo que seu servidor esteja listado (local é o servidor SQL Server local), clique no servidor que você quer registrar e clique em Add. Isso move o servidor SQL Server para o lado direito da janela, embaixo de Servidores Adicionados [Added Servers]. Você pode registrar mais de um servidor de uma vez se você quiser.

5. Depois que você tiver adicionado os servidores SQL Server desejados, clique em Next, e vai aparecer uma janela perguntando a você qual opção de conexão que você quer utilizar para se conectar ao SQL Server.

Você tem duas opções: autenticação do Windows NT ou autenticação do SQL Server. Se esta é a primeira instalação do SQL Server na sua organização, escolha por agora autenticação do SQL Server; isso pode ser mudado mais tarde se você mudar seu modo de segurança. Mas se não for a primeira instalação do SQL Server, escolha o modo de segurança que os outros servidores SQL Server estiverem utilizando. O exemplo a seguir supõe que você



tenha escolhido a autenticação do SQL Server [SQL Server authentication]. Depois de feita sua escolha, clique em Next.

6. A seguir, o assistente quer que você escolha se o SQL Enterprise Manager se lembre do seu nome de login e senha.

A primeira opção, "Efetuar logon automaticamente usando minha informação de conta do SQL Server" [Login Automatically Using My SQL Server Account Information], pode ser escolhida para que o Enterprise Manager se lembre de sua senha e nome de login. Assim, você não precisa de, a cada vez que iniciar o Enterprise Manager, ficar digitando essa informação novamente.

Se você não quiser que o Enterprise Manager lembre-se de seu login e senha, escolha a opção "Perguntar a informação da conta do SQL Server quando se conectar" [Prompt for the SQL Server account information when connecting].

Se você escolher a primeira opção, digite SA como nome de login, e deixe a senha [Password] em branco. Você deve usar esse nome de login e senha quando registra pela primeira vez um novo servidor SQL Server porque você ainda não atribuiu à conta SA uma senha, nem criou qualquer outro nome de login. Clique em Next para continuar.

7. Você deve adicionar um servidor SQL Server a um grupo de servidores.

Nessa tela, você pode adicionar ao grupo padrão SQL Server Group, ou a um outro grupo preexistente (selecionando a primeira opção). Você pode ainda criar um novo grupo (selecionando a segunda opção [Create a new top-level SQL Server group to add the SQL Server(s) to.]), e seu servidor será adicionado a esse grupo. Grupos de servidores são usados para agrupar servidores SQL Server de usos semelhantes, para fins de administração, e são completamente opcionais. Tudo que os grupos de servidores fazem é agrupar grupos de servidores semelhantes para visualização no Enterprise Manager. Escolha a opção mais adequada e clique em Next para continuar.

The screenshot shows a Windows-style dialog box titled "Register SQL Server Wizard". The main heading is "Select Connection Option". Below this, a subtitle reads: "When you connect using SQL Server account information you can store your login name and password or provide it each time you connect." There are two radio button options. The first option, "Login automatically using my SQL Server account information.", is selected. Below it are two text input fields: "Login name:" with the text "sa" entered, and "Password:" which is empty. The second option, "Prompt for the SQL Server account information when connecting", is unselected. At the bottom of the dialog are three buttons: "< Back", "Next >", and "Cancel".

8. O assistente de registro mostra sua última tela.

Se você quiser fazer quaisquer mudanças, você pode fazê-las clicando no botão Back. Ou, se tudo estiver conforme você queria, clique em Finish para concluir. A caixa de diálogo de registro do SQL Server aparece, e você recebe uma mensagem dizendo se o registro foi bem-sucedido. Se você receber uma mensagem de erro, veja a seção de solução de problemas.



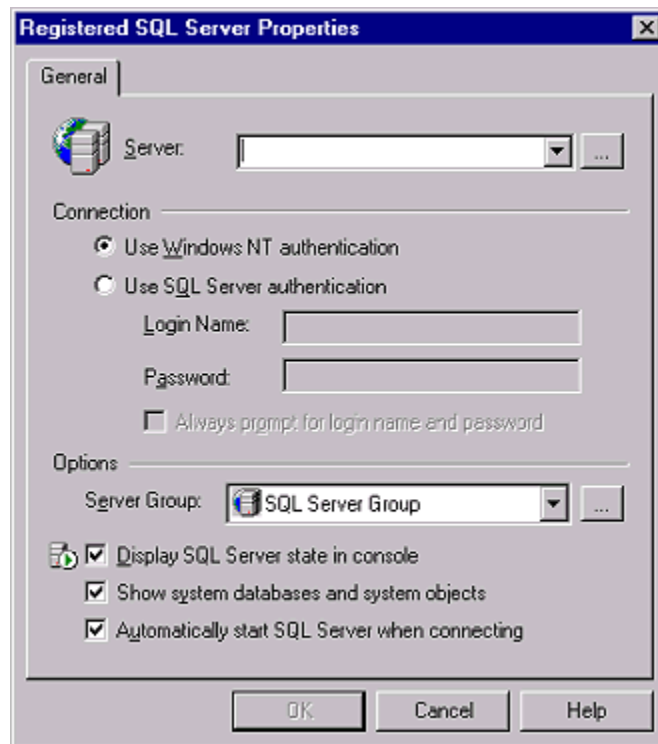
9. Clique em Close, e volta-se para o SQL Enterprise Manager. Embaixo de Console Root, você verá o cabeçalho Microsoft SQL Servers. Clique no sinal de mais perto deste cabeçalho, e um ou mais grupos de servidores serão mostrados. O(s) servidor(es) que você acabou de registrar aparecerão dentro do grupo que você definiu para ele(s). Supondo que você não tenha criado nenhum grupo e tenha adicionado-os ao grupo padrão, o único grupo listado será o padrão SQL Server Group. Para ver o servidor que você acabou de registrar, clique no sinal de mais perto do grupo de servidores em que você adicionou o(s) servidor(es). Se o servidor não aparecer dentro de nenhum grupo de servidores, ele não foi registrado corretamente com o Enterprise Manager.

Registrando um servidor manualmente

Se você não quiser usar o Assistente de Registro, você não precisa. Registrando um servidor manualmente com o Enterprise Manager



1. O primeiro passo é desligar o Assistente de registro. Depois que ele estiver desligado, então dessa vez em diante, você será capaz de registrar manualmente qualquer servidor com o Enterprise Manager. Para desligar o assistente de registro, você deve primeiro iniciá-lo clicando com o botão direito no texto Microsoft SQL Server, ou em qualquer grupo de servidores, ou em qualquer servidor registrado, e então selecionar a opção Novo registro de servidor [New SQL Server Registration]. Isso mostra a primeira tela do Assistente de registro.
2. Para desligá-lo, selecione a opção "A partir de agora eu quero realizar essa tarefa sem usar um assistente" [From now on I want to perform this task without using a wizard]. Clique em Next e na outra janela em Cancel. Isso fechará o assistente de registro e também o impedirá de ser executado a cada vez que você for registrar um servidor.
3. Agora, para registrar um servidor manualmente, clique com o botão direito no texto Microsoft SQL Server, ou em qualquer grupo de servidores, ou em qualquer servidor registrado, e então selecionar a opção Novo registro de servidor [New SQL Server Registration]. Isso mostra a caixa de diálogo "Propriedades do servidor SQL registrado" [Registered SQL Server Properties]



4. Complete essa janela usando o mesmo tipo de informações que foram descritas quando descrevemos o Assistente de registro. Uma diferença entre o Assistente de registro e essa caixa de diálogo são as três opções na parte de baixo da janela.

Display SQL Server State in console - Esta opção, se selecionada, faz com que o Enterprise Manager regularmente interroge o serviço MSSQLServer para saber se ele está rodando, e mostra uma luz verde no ícone no console quando ele está rodando, e uma luz vermelha se o serviço não estiver sendo executado.

Show System Databases and System Objects - Se esta opção for escolhida, todos os bancos de dados e objetos do sistema serão mostrados no console.

Caso não seja escolhida, os mesmos não aparecem no console.

Automatically Start SQL Server when connecting - Assumindo que o serviço MSSQLServer não esteja ajustado para iniciar automaticamente, esta opção pode ser usada para iniciar automaticamente este serviço quando da primeira conexão com o servidor.

Por padrão, as três opções estão escolhidas agora, e são escolhidas automaticamente quando se usa o assistente de registro. Você pode fazer qualquer escolha que achar adequada. Quando você tiver completado suas escolhas, você pode registrar o servidor clicando em OK.

Como editar as informações de registro do servidor SQL Server

Às vezes, você pode precisar de editar as configurações de registro do SQL Server, como quando você mudar o login ou a senha que você usou originalmente para registrar o servidor.

Fazendo mudanças em um servidor registrado

1. No Enterprise Manager, selecione o servidor cujo registro você quer alterar.
2. Clique com o botão direito no nome do servidor e então escolha Edit SQL Server Registration no menu. Isso mostra a caixa de diálogo Registered SQL Server Properties, que foi mostrada na figura acima.
3. Faça quaisquer mudanças necessárias. Quando terminar, clique em OK para salvar essas configurações e voltar para o Enterprise Manager

Como cancelar o registro de um servidor

De tempos em tempos, pode ser necessário cancelar o registro de um servidor, no Enterprise Manager. Para isso, faça:

1. No Enterprise Manager, selecione o nome do servidor cujo registro você quer cancelado.
2. Clique com o botão direito em seu nome, e selecione Delete do menu.
3. Uma caixa de confirmação aparece, pedindo-lhe para clicar em Yes para remover o servidor, ou No para cancelar a operação. Clique em Yes para cancelar o registro do servidor.

Gerenciando grupos de servidores

Embora você possa criar novos grupos a partir do assistente de registro, ou a partir da caixa de diálogo Registered SQL Server Properties, você também pode criar, renomear e excluir grupos de servidores manualmente. Você também pode mudar um servidor de um grupo para outro se você quiser. Para gerenciar os grupos de servidores, faça o seguinte:



1. A partir do Enterprise Manager, clique no sinal de mais perto do cabeçalho Microsoft SQL Servers. Isso mostra todos os grupos de servidores atualmente embaixo dele. Clique com o botão direito no nome Microsoft SQL Server, ou em qualquer grupo de servidores, e então selecione Novo grupo de servidores [New SQL Server Group] do menu. Aparece a caixa de diálogo de Grupos de Servidores.
2. Para criar um novo grupo de servidores, entre com o nome do novo grupo na caixa Name e clique em OK. Isso fará esse grupo automaticamente um grupo de servidores do nível mais alto. Se você quiser, você pode escolher criar um grupo como subgrupo de algum outro grupo, escolhendo a opção "Sub-grupo de" [Sub-group of:]. Na maioria dos casos, grupos de servidores do mais alto nível são mais do que suficientes.
3. Se você quiser remover um grupo de servidores, clique com o botão direito no nome do grupo de servidores que você quer remover, e então selecione Delete do menu. O grupo é removido imediatamente sem nenhum aviso.

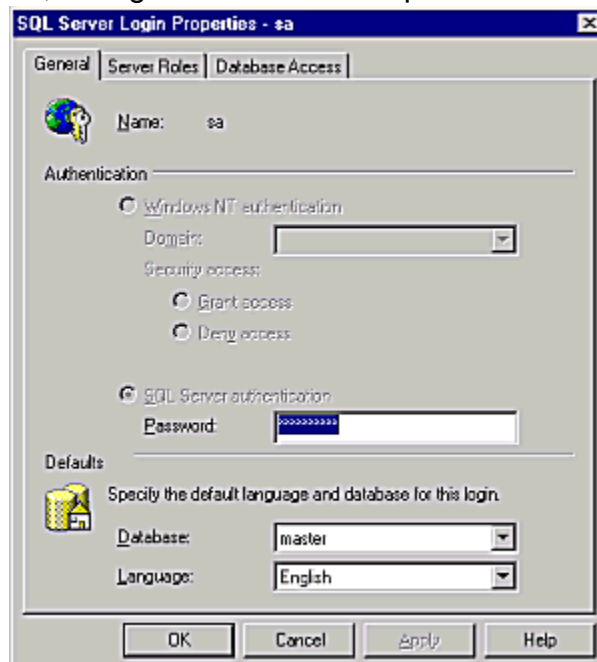
Atribuindo uma senha ao Administrador do Sistema (SA)

Se você estiver usando o método de autenticação do NT, você não precisa executar este passo porque não se exige que você efetue logon no servidor SQL Server diretamente usando um nome de login [login ID].

Mas se você estiver usando o método de autenticação do SQL Server, uma das primeiras coisas que você vai querer fazer é definir uma senha para a conta SA. O SA tem a capacidade de executar qualquer função no SQL Server, e você deve evitar que usuários não autorizados efetuem logon no servidor SQL Server.

Colocando uma senha para a conta SA

1. Do Enterprise Manager, abra a pasta que representa o servidor SQL Server, cuja senha do SA precisa ser alterada.
2. Abra a pasta Security [Segurança], e clique uma vez em Logins, e então no lado direito da janela, os logins atualmente disponíveis são exibidos.



3. Clique com o botão direito no nome de login SA, e selecione Propriedades [Properties] no menu; a caixa de diálogo Propriedades aparece.
4. Para adicionar uma senha para a conta SA, digite-a na caixa identificada por Password. Você vai notar que a caixa da senha está preenchida com asteriscos. Isso não significa nada, já que atualmente não há senha para o login SA. Escolha uma senha que não seja fácil de adivinhar.
5. Depois que você tiver colocado uma senha, clique em OK. Isso salvará a senha e fechará a caixa de diálogo. Agora você precisa de voltar ao registro desse servidor e editá-lo para indicar a nova senha da conta SA. (para editar o registro do servidor, faça como indicado em Registrando um servidor manualmente)

O restante desta tela de Propriedades será visto no decorrer do curso.

Agora, o SQL Server está funcionando e pronto para ser configurado para qualquer aplicação na qual ele esteja sendo utilizado.

Solução de problemas de instalação

Se você seguiu corretamente os avisos e instruções até aqui, você não deve encontrar problemas na instalação do SQL Server. Quase todos os problemas que você encontrar são porque você deixou passar uma etapa, ou cometeu algum engano durante o processo de instalação. Aqui, vamos comentar alguns problemas comuns de instalação e como corrigi-los.

Como identificar problemas na instalação do SQL Server

Há muitas maneiras de identificar que sua instalação do SQL Server falhou. Entre eles:

- Mensagens de erro. A resposta mais comum que o SQL Server te fornece são mensagens de erro. Embora as mensagens que você receba possam não ser sempre precisas, a primeira suposição que se deve fazer quando receber uma mensagem de erro é que ela é precisa e que você precisa encontrar a causa dela. Algumas mensagens são óbvias, já outras são bem obscuras. Se você não puder determinar a partir da própria mensagem, qual é o problema, escreva a mensagem por inteiro e procure nos livros on-line do SQL Server (SQL Server books on-line). Se lá você não encontrar a mensagem, tente olhar na Microsoft TechNet (você tem que ser um assinante para recebê-la) ou no site da Microsoft na Internet.

Se você não encontrar a mensagem de erro, você pode checar o Visualizador de Eventos [Event Viewer] do Windows NT, no log de aplicativo, para alguma mensagem relacionada. Você ainda pode ver os logs de erro do SQL Server, se houver algum, para pistas. Os logs de erro do SQL Server estão localizados na pasta \log abaixo da pasta onde o SQL Server foi instalado. Encontre o arquivo chamado Errorlog e abra-o com o Notepad ou Wordpad.

Há também o arquivo Sqlstp.log, na pasta C:\WINNT, que lhe dá informações sobre o processo de instalação do SQL Server.

Todos os logs de erros são arquivos ASCII e podem ser facilmente visualizados com qualquer editor de textos. Os logs de erros do SQL Server são difíceis de interpretar, mas podem te dar uma dica do que aconteceu.

- Não foi possível verificar a instalação. Se você tentou verificar a instalação como descrito em Verificando a instalação, e o problema parece ser algo faltando da instalação que deveria estar aí, você tem duas opções. Ou execute o programa de instalação novamente sobre a instalação atual, ou você pode primeiro excluir a instalação danificada, e então reinstalar.
- Verificar o arquivo Cnfgsvr.out. Esse arquivo, localizado na pasta \INSTALL, abaixo da pasta onde o SQL Server foi instalado, é um arquivo de saída gerado pelos scripts que rodam durante a instalação e grava mensagens de erro DBCC.

A seguir são descritos alguns dos problemas mais comuns que são encontrados quando da instalação do SQL Server.

Uso da versão errada do NT Server

O uso da versão errada do NT Server, ou o uso de um Service Pack antigo, pode causar uma grande variedade de problemas, muito deles difíceis de diagnosticar. Se a instalação produz mensagens de erro obscuras sem razão aparente, assegure-se de que a versão do NT Server que você está usando seja adequada. Se você descobrir que tem a versão errada, desinstale o SQL Server, atualize o NT Server para uma versão aceitável, e então reinstale o SQL Server.

Arquivos abertos durante a instalação

Durante o processo de instalação, o SQL Server substitui alguns arquivos do NT Server. Se algum desses arquivos estiver aberto durante a instalação, eles podem causar a exibição de uma mensagem de erro crítico. Essa é a razão de ser importante certificar-se de que não há nenhum outro programa rodando quando o SQL Server for instalado. Se você descobrir que tem um ou mais utilitários do NT aberto que pode estar causando o erro, feche-os, e clique no botão de Retry mostrado pela mensagem de erro. Se isso não funcionar, você pode ter que abortar a instalação do SQL Server e tentar de novo, desta vez sem nenhum programa rodando.

Os serviços MSSQLServer ou SQLServerAgent não iniciam

Este é provavelmente o problema mais comum encontrado quando se instala o SQL Server. Siga os passos abaixo para te ajudar a determinar a possível causa desse problema:

- Você criou uma conta de serviço como descrito na instalação do software de servidor?
- A conta de serviço foi criada adequadamente, com direitos administrativos e os outros direitos avançados exigidos?
- A conta de serviço foi criada no domínio de contas correto do NT?
- A conta de serviço foi informada corretamente quando pedida durante o processo de instalação? Você usou acidentalmente sua conta de logon no NT como a conta de serviço?
- Você digitou tudo corretamente?

Se você não conseguir descobrir o problema, delete a conta de serviço que você criou e crie uma nova, seguindo cuidadosamente as recomendações da instalação do software de servidor. Então vá para o Painel de Controle, onde estão os serviços, e assegure-se de que você selecionou a conta de serviço para os dois serviços, junto com as senhas corretas. Com frequência, esse é um erro simples que é facilmente corrigido.

3 - Ferramentas de gerenciamento do SQL Server

MMC - Microsoft Management Console

Enterprise Manager

Service Manager

Client Network Utility

Server Network Utility

Performance Monitor

Pofiler

Query Analyzer

Books online

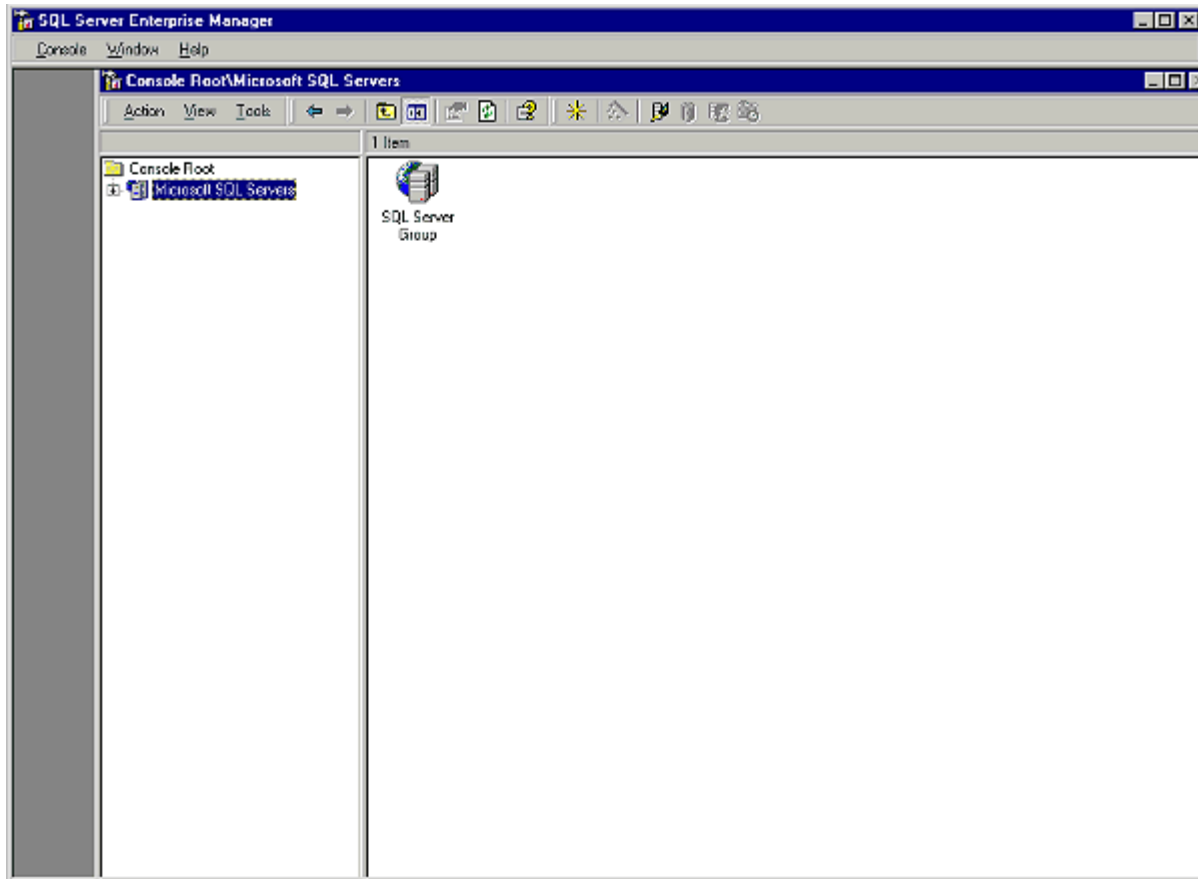
Assistentes do SQL Server

Objetivos:

- Conhecer os conceitos de alocação de espaço usados pelo SQL Server;
- Ter uma visão geral dos itens que compõem o catálogo do sistema;
- Saber o que é um banco de dados e o que ele contém;
- Aprender a criar, usar e gerenciar dispositivos de banco de dados.

SQL Server Enterprise Manager

O "SQL Server Enterprise Manager" é a porta de entrada para a Interface de usuário do SQL Server. Para iniciá-lo, selecione Iniciar, Programas, Microsoft SQL Server 7.0, Enterprise Manager. Aparece o Enterprise Manager dentro do MMC, como abaixo.



Aí temos uma porção de menus e botões. Os itens de menu importantes são Action, View e Tools:

- Action te permite fazer coisas tais como registrar um novo servidor ou um novo grupo (conforme visto na seção de instalação).
- Views te fornece uma lista dos diferentes tipos de visões disponíveis para você. Você pode selecionar as visões grande, pequeno, detalhe ou lista dos ícones e suas propriedades associadas. Ainda é possível definir quais itens e quais barras de ferramentas você verá. Exatamente como no Windows Explorer.

- O menu Tools lista todas as ferramentas e assistentes do SQL Server. Você pode fazer backup de um banco de dados; parar, iniciar e configurar a replicação; e iniciar ferramentas como o Query Analyzer (Analisador de consultas), entre outras.

Imediatamente à direita dos menus há outros itens de barras de ferramentas. Estes são basicamente atalhos para os itens mais usados da barra de menu. Temos, entre outros:

Anterior



Próximo



Atualizar



Registrar Servidor



Novo Banco de Dados



Novo Login



Query Designer


O SQL Server 7.0 tem uma ferramenta muito útil, que se parece com o Query By Example (QBE) do Microsoft Access, e é um ótimo substituto para a MS Query (do SQL Server 6.5). O nome dessa aplicação é Query Designer e faz parte das Ferramentas Visuais de Banco de Dados. É uma ótima ferramenta, mas não está listada como uma ferramenta do SQL Server, e é um pouco difícil achá-la diretamente. Para localizá-la, faça assim:

- Do Enterprise Manager, expanda o banco de dados Northwind, e expanda as tabelas.
- Clique com o botão direito na tabela Categories.
- Selecione Open Table, e então Return All Rows.

Nota: O que foi citado acima para a tabela Categories, do banco de dados Northwind, aplica-se a qualquer tabela, de qualquer banco de dados, assim como não é

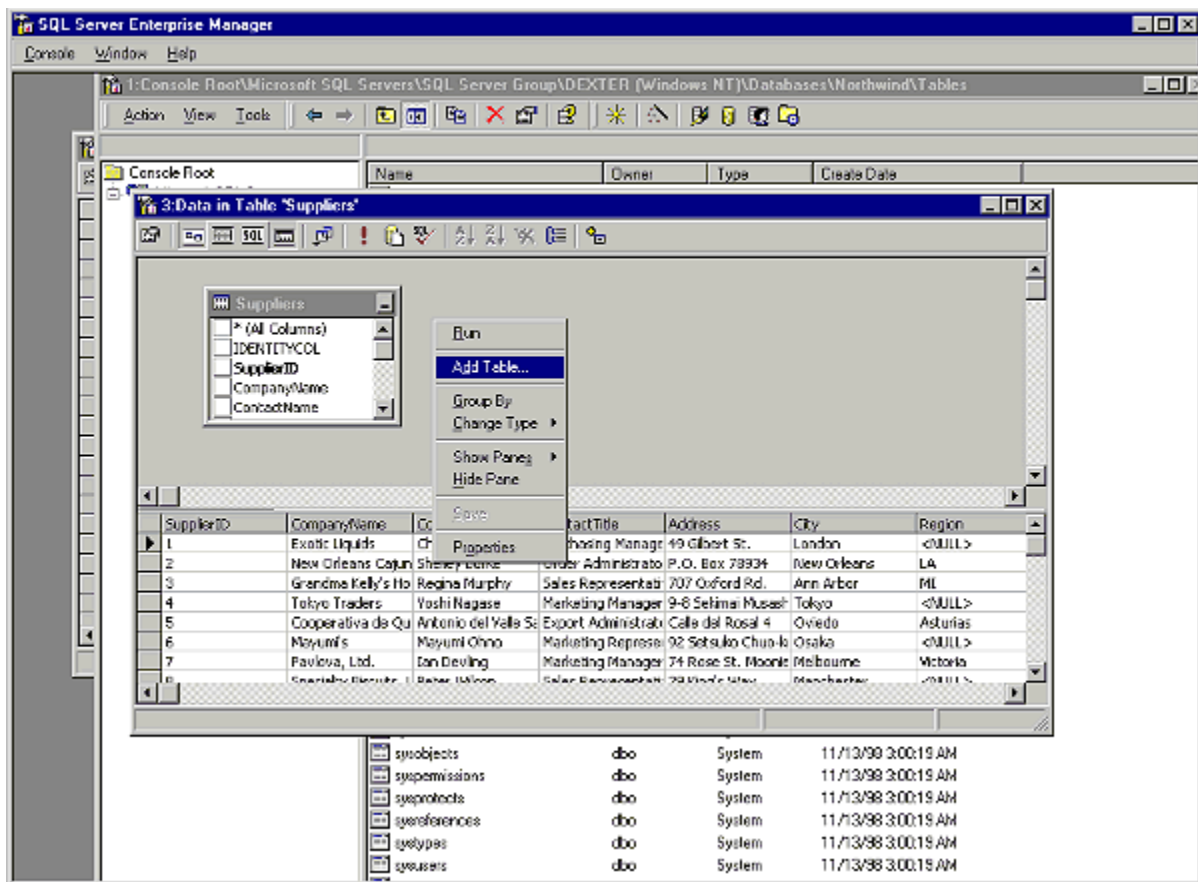
necessário selecionar Return All Rows. Poderia ter sido selecionado Return Top... que te perguntaria quantos registros você quer ver. A diferença é no comando SQL gerado para cada uma das seleções.

Depois que você abrir a tabela, toda a janela estará envolvida com o Query Designer. Ele permite que você veja propriedades da consulta, execute consultas de seleção, de ação (inserção, atualização, exclusão, e criação de tabelas), validação da sintaxe SQL, ordenação, filtragem, e agrupamentos.

Provavelmente, estaremos vendo a seção dos Resultados (Results Pane), acionada com o botão  , que mostra os resultados atuais da consulta. Porém, ainda há outras três seções para se escolher ou adicionar à janela atual. Estas são, Show Diagram, Show Grid, e Show SQL Pane. Elas são selecionadas pelos botões no topo da janela.

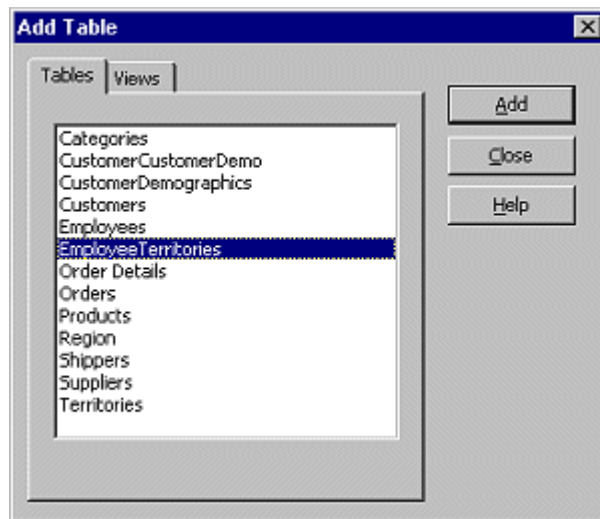
Show Diagram Pane

Esta seção te permite visualizar graficamente o banco de dados ou as tabelas. Por

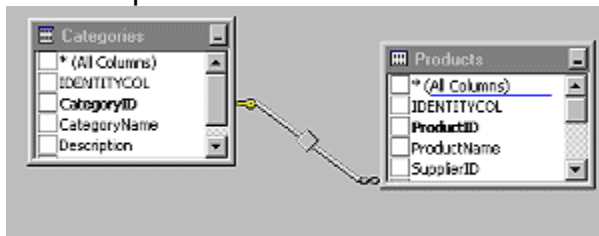


padrão, só a tabela atual que estiver aberta será mostrada nesta seção. Para adicionar outra tabela, simplesmente clique com o botão direito em qualquer lugar da seção (exceto na representação gráfica de alguma tabela). Aparece um menu com algumas opções.

Selecione Add Table. Isso mostrará todas as tabelas e visões (views) disponíveis no seu banco de dados, conforme abaixo.



Adicione a tabela Products (supondo que você seguiu o exemplo acima e está vendo as tabelas do banco de dados Northwind), selecionando-a e clicando em Add, ou dando um duplo clique no nome da tabela. Note que o relacionamento também é representado graficamente com uma linha e chave em uma ponta e um ícone de infinito na outra ponta.




Isso mostra um relacionamento de um para muitos(1:N) com CategoryID na tabela Categories como a chave primária. Pode-se selecionar para a consulta, colunas individuais ou todas as colunas marcando as caixas de verificação apropriadas perto do nome de cada coluna.

Show Grid Pane

Esta seção permite detalhar mais a consulta sendo criada. Por exemplo, podemos selecionar a ordem para uma coluna particular, seu alias (nome que será mostrado no grid). Tudo isso de uma maneira visual.

Show SQL Pane

Esta seção mostra a declaração SQL gerada pelas seleções feitas nas seções acima. Pode-se agora copiar e colar código desta janela, para o Query Analyzer, pra um ambiente de programação, entre outros.

Nota: A seção de resultados (Results Pane) não se atualiza automaticamente. Para atualizá-la, clique no ponto de exclamação vermelho (Run) 

No Query Designer, podemos criar consultas complexas muito mais rapidamente que no Query Analyser (que será visto adiante). Depois de criarmos as consultas, podemos colocá-las no Query Analyzer e testar sua performance.

SQL Server Client Network Utility

A ferramenta SQL Server Client Configuration é utilizada para configurar as ferramentas de gerenciamento, de modo que elas possam comunicar-se com sucesso com um servidor SQL Server.

A ferramenta SQL Server Client Configuration Utility está localizada no grupo de programas do SQL Server 7.0 (Iniciar | Programas | Microsoft SQL Server 7.0 | Client Network Utility). Essa ferramenta é chamada de *Client Configuruation Utility* e de *Network Configuration* nos livros on-line.

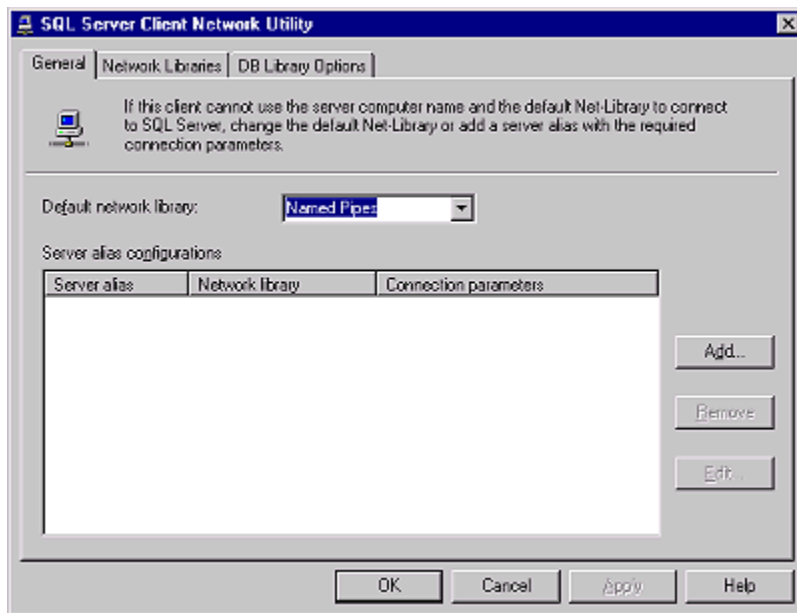
Na maioria das situações, você não precisará de executar este programa porque as configurações padrão que foram utilizadas durante a instalação vão funcionar na maioria das configurações de estações de trabalho. Mas se você descobrir que as suas ferramentas de cliente não conseguem se comunicar com o servidor SQL Server, você pode utilizar esse utilitário para configurar adequadamente o cliente de modo que ele possa comunicar-se com o servidor SQL Server.

Iniciando a Client Network Utility

Provavelmente, esta ferramenta foi instalada quando você instalou as outras ferramentas de gerenciamento em uma estação de trabalho. Se você não instalou esta ferramenta, você deve instalá-la seguindo os passos descritos em Instalando as ferramentas de cliente. Uma vez que a Client Netowrk Utility esteja instalada, execute-a seguindo os passos abaixo:

1. Efetue login na estação de trabalho usando uma conta com as permissões adequadas.
2. Para executar a Client Network Utility, clique em Iniciar | Programas | Microsoft SQL Server 7.0 | Client Network Utility.

3. Aparece a janela da Client Network Utility.



4. Essa ferramenta inclui três guias que separam cada uma das três opções principais. Elas são General, Netowrk-Libraries, e DB-Library Options. A seguir descreveremos cada uma delas.

Geral

A guia geral (figura acima) tem duas seções. Na parte superior da tela você pode especificar qual Net-Library você quer usar como o protocolo padrão de rede para este cliente. A segunda parte da tela é utilizada para especificar configurações opcionais do protocolo de rede/

Como você deve se lembrar, quando o SQL Server foi instalado, você teve que especificar uma ou mais Net-Libraries para serem instaladas. Este é o software utilizado para estabelecer uma conexão de rede entre o servidor SQL Server e o

software de cliente. O SQL Server usa as bibliotecas de rede [Net-Libraries] para se comunicar com um protocolo de rede específico e enviar pacotes através da rede entre um cliente e um servidor. O servidor escuta simultaneamente em diversas portas, enquanto o cliente se comunica com o servidor usando uma Net-Library específica. Para que um cliente se conecte a um servidor, ele deve usar alguma das Net-Libraries que o servidor tem instaladas.

Você pode fazer com que cliente e servidor usem a mesma Net-Library de duas maneiras: adicionar no cliente, uma Net-Library que está instalada no servidor; ou o contrário: no cliente, instalar uma Net-Library que esteja instalada no servidor.

Normalmente, é mais fácil adicionar a Net-Library ao servidor, do que aos clientes. A configuração de uma Net-Library no cliente, para se comunicar com um servidor, é opcional. Por padrão, Named Pipes é a Net-Library dos clientes, instalada durante a instalação do SQL Server para computadores executando Windows NT ou Windows 9x. Named Pipes deve funcionar bem na maioria dos casos. Mas, se esse protocolo não conectar-se com seu servidor, você vai precisar de reconfigurar o cliente com a configuração correta.

Então, caso você precise mudar o protocolo de rede para uma estação de trabalho (cliente), você pode fazer isso selecionando o protocolo apropriado da lista Protocolo de rede padrão [Default network Protocol].

A Net-Library de cliente padrão para os clientes SQL Server fazendo conexões remotas é Named Pipes, a qual não é suportada em servidores rodando Windows 9x. Clientes conectando-se com servidores rodando Windows 9x devem usar a ferramenta SQL Server Client Network Utility para executar uma das seguintes opções:

- Mudar a Net-Library padrão do cliente.
- Definir uma entrada de configuração para uma Net-Library de cliente na qual o servidor rodando Windows 9x esteja escutando.

A parte de configuração do protocolo de rede da guia Geral [General] só é usada em casos especiais. Você só vai utilizá-la se se ver em uma das seguintes situações:

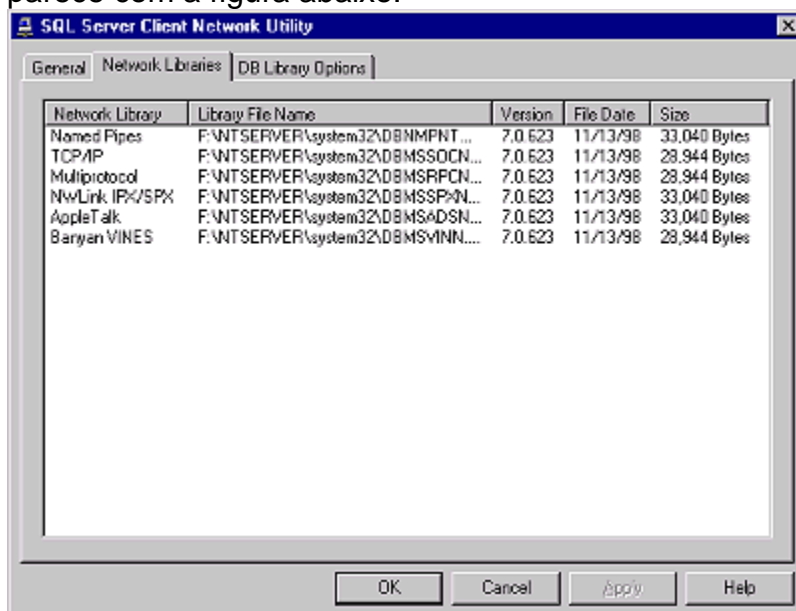
- As ferramentas de gerenciamento estão em uma estação de trabalho rodando Windows NT, conectando-se a um servidor executando Windows 95.
- Você precisa adicionar uma configuração exclusiva de protocolo para as comunicações entre um servidor SQL Server específico e um cliente executando as ferramentas de gerenciamento.
- O servidor SQL Server com o qual você quer se comunicar a partir de um cliente escuta em uma porta não-padrão.

Normalmente, você não usará esta opção.

Nota: Para o processo Servidor, o SQL Server escuta as Net-Libraries Named Pipes, sockets TCP/IP, e Multiprotocolo em computadores rodando Windows NT. Entretanto, Named Pipes não é aceito em computadores rodando Windows 9x. O SQL Server instalado em computadores rodando Windows 9x escuta as Net-Libraries sockets TCP/IP e Multiprotocolo do servidor. Se a conexão é local com o servidor (tal como um cliente e servidor na mesma máquina), o SQL Server vai escutar então a Net-Library de Memória Compartilhada do servidor.

Network Libraries

A guia Network Libraries é usada somente para mostrar quais Network Libraries estão atualmente instaladas no cliente e permitir que você saiba suas versões. Essa guia se parece com a figura abaixo.



DB-Library Options

O principal objetivo da guia DB-Library é lhe permitir determinar se você tem ou não a versão mais atual dos arquivos de DB-Library instalados no cliente.

Outra parte da guia DB-Library Options é formada pelas duas caixas de verificação que podem ser usadas para configurar como a DB-Library se comunica com o SQL Server. Eis o que elas fazem:

- **Automatic ANSI to OEM:** Quando esta opção estiver selecionada, a DB-Library converte caracteres do formato OEM para ANSI quando ocorre a comunicação do cliente para o servidor SQL Server, e converte caracteres do formato ANSI para OEM quando se comunica do servidor para o cliente. Esta opção é exigida frequentemente porque o conjunto de caracteres utilizado pelo SQL Server é diferente daquele utilizado pelo sistema operacional do cliente. Esta opção faz automaticamente a tradução adequada entre os dois conjuntos de caracteres. Essa opção deve estar selecionada se o cliente estiver executando Windows NT ou Windows 9x. Se estiver sendo utilizado um cliente Windows 3.x, esta opção não deve ser selecionada.
- **Use International Settings:** Quando esta opção estiver selecionada, permite-se que a DB-Library pegue as configurações de formato de data, hora e moeda do sistema operacional local ao invés de utilizar a configuração definida no código do SQL Server. Esta opção deve ser selecionada se o cliente estiver sendo executado no Windows 9x ou NT.

SQL Server Profiler

O SQL Server Profiler é uma ótima ferramenta para se ver um registro contínuo da atividade do servidor em tempo real. O Profiler monitora os eventos produzidos através do SQL Server, filtra esses eventos baseados em critérios específicos do usuário, e mostra a saída traçada na tela, em um arquivo ou uma tabela.. Você pode até repetir traçados capturados anteriormente.

Monitorando com o SQL Server Profiler

O SQL Server Profiler é uma ferramenta gráfica que permite aos administradores do sistema monitor eventos de mecanismo do SQL Server. Eventos são a nova maneira de se comunicar com o SQL Server. Com eventos de mecanismo, um objeto COM pode interceptar esses eventos e agir de acordo. Exemplos de eventos de mecanismo incluem:

- Comandos Transact-SQL: SELECT, INSERT, UPDATE e DELETE.
- Conexão, falha ou desconexão de login.
- O começo ou fim de um procedimento armazenado (stored procedure).
- O começo ou fim de um lote de comandos SQL.
- Um erro escrito no log de erros do SQL Server.
- Um bloqueio adquirido ou liberado em um objeto de banco de dados.
- Um cursor que for aberto

Os dados gerados sobre cada evento podem ser capturados e salvos em um arquivo ou uma tabela do SQL Server para análise posterior. Para coletar os dados dos eventos de mecanismo você define traços. Exemplos de dados capturados com um traço incluem:

- O tipo (classe) de um evento, tal como Object:Created, o qual indica que um objeto de banco de dados foi criado.

- O nome do computador em que o cliente está rodando.
- O ID do objeto afetado pelo evento, tal como um nome de tabela.
- O nome no SQL Server, do usuário que executou o comando.
- O texto do comando Transact-SQL ou procedimento armazenado.
- A hora em que o evento começou e terminou.

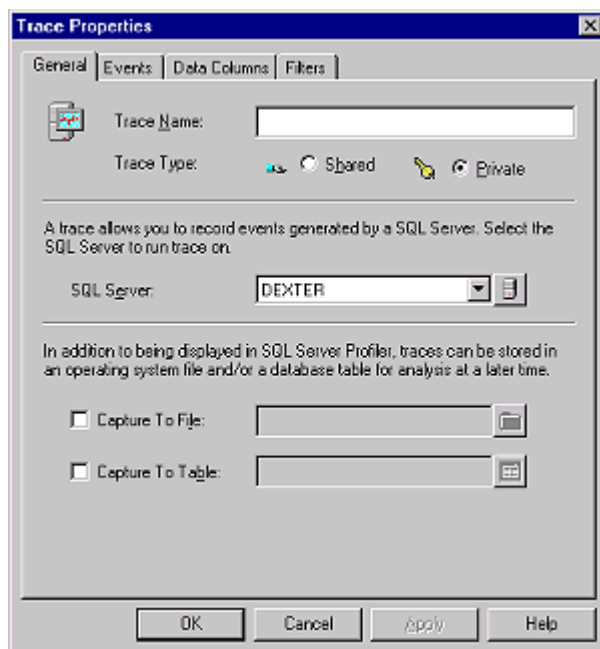
Você pode filtrar os dados de modo que apenas um subconjunto dos dados do evento seja coletado. Isso te permite coletar apenas os dados de evento em que você está interessado. Por exemplo, se você apenas está interessado em um usuário particular ou nos efeitos de um certo banco de dados, você pode filtrar esses objetos específicos e ignorar os outros. Você também pode definir filtros em itens que demorem mais do que o esperado, tal como uma consulta que demore mais do que 40 segundos.

O SQL Server Profiler também permite que os dados de eventos (capturados) sejam recolocados no SQL Server, Isso vai efetivamente reexecutar os eventos salvos como eles originalmente ocorreram.

O SQL Server Profiler pode ser usado para:

- Monitorar o desempenho do SQL Server.
- Depurar comandos Transact-SQL e procedimentos armazenados.
- Identificar consultas que executam com lentidão.
- Corrigir problemas no SQL Server. Por exemplo, você pode capturar os eventos que possam estar levando a um problema potencial e então replicar o processo em um sistema de teste para isolar e corrigir o problema.

Abaixo você vê a caixa de diálogo inicial para a criação de um traço. Veremos a criação de traços em mais detalhes posteriormente.



SQL Server Query Analyzer

O Analizador de Consultas [Query Analyzer] fornece uma interface gráfica para analisar o plano de execução de uma ou múltiplas consultas, ver os dados de resultado, e recomendar índices .

O Query Analyzer se parece com a janela Query que havia no Enterprise Manager do SQL Server 6.5. Se você já for familiarizado com a versão 6.5, essa ferramenta será bem fácil de se usar; mas, de qualquer maneira, aqui será explicado como utilizá-la. Essa é uma ferramenta bem fácil de se usar. Vamos começar pela execução de uma consulta simples.

Usando o Query Analyzer

1. A partir do MMC, selecione Tools | Query Analyzer. Você também pode executar o Query Analyzer pelo grupo de programas do SQL Server (Iniciar | Programas | Microsoft SQL Server 7.0 | Query Analyzer).
2. Conecte-se ao servidor SQL Server local. Efetue login com a conta e senha (se houver) do SA que você especificou na instalação. Caso você tenha privilégios administrativos nessa máquina, você também pode se logar com a conta do NT [use Windows NT authentication]. Dessa maneira, você também efetuará login como SA.
3. Selecione pubs da lista na janela de consulta. Esse será o banco de dados no qual executaremos a consulta.
Você também poderia ao invés de especificar o banco de dados, usar a declaração Use antes de sua consulta.
4. Escreva o seguinte na parte superior da janela se a janela tiver mais de uma parte.

```
SELECT * FROM authors
```
5. Selecione o método de execução. São disponíveis quatro opções diferentes de execução:
Execução padrão (Ctrl+T) mostra os resultados na forma de um arquivo texto, como na figura abaixo. F5 (run) executa a consulta, qualquer que seja o modo de execução selecionado.
Executar para uma grade (Ctrl+D) fornece um formato mais agradável de se visualizar, em uma planilha, uma estrutura parecida com tabelas (um layout de linha-coluna). A figura abaixo mostra o resultado em uma grade.

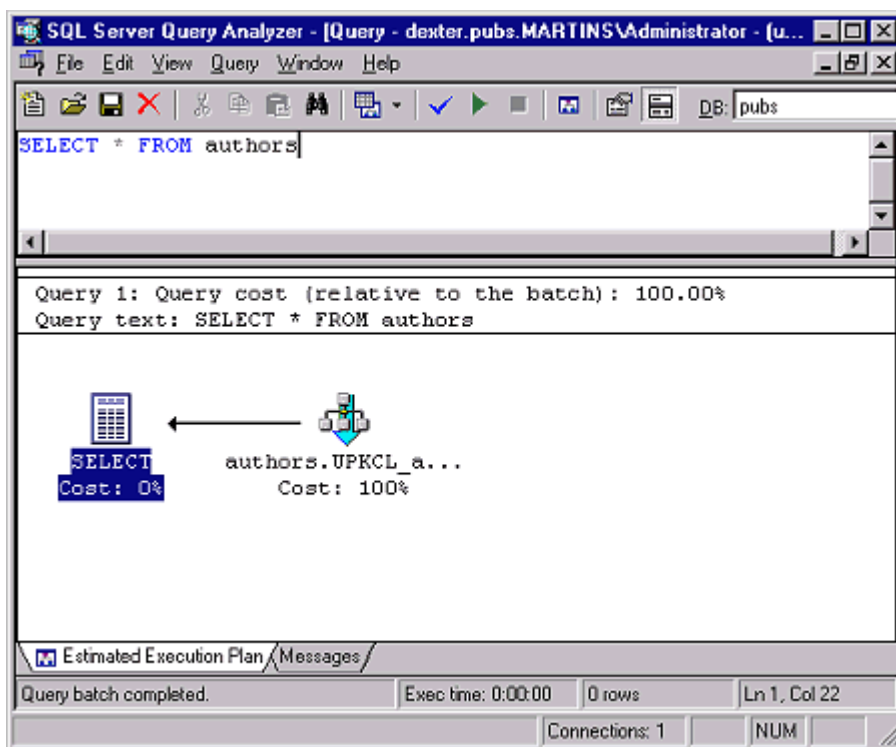
Plano de execução (Ctrl+L), mostrado na próxima figura, mostra o processo real de execução que ocorreu. Essa consulta não mostrou nada de interessante porque só havia uma tabela envolvida, mas se fosse executada uma consulta mais complexa com junções internas ou externas, você visualizaria sua representação gráfica.

Análise de Índice [Index Analysis] provavelmente lhe avisará que ele foi incapaz de recomendar quaisquer índices. Entretanto, se você começar a escrever consultas complexas que são usadas frequentemente, você pode colá-las nesse utilitário e testá-las para sugestões de índice.

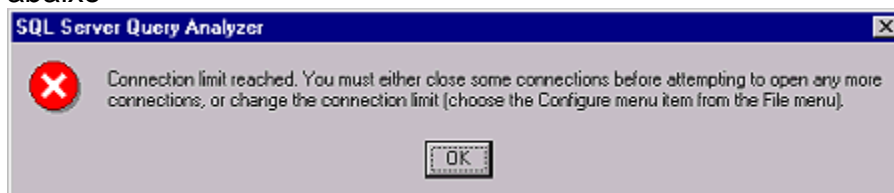
Quando você decide fazer uma nova consulta, você pode fazê-la em outra janela, clicando no botão New Query (Ctrl+N), ou então apagar a consulta já feita e digitar novos comandos.



Note que a cada nova janela aberta, ou seja, a cada consulta feita em outra janela, está sendo estabelecida uma nova conexão. Observe na parte inferior da janela do Query



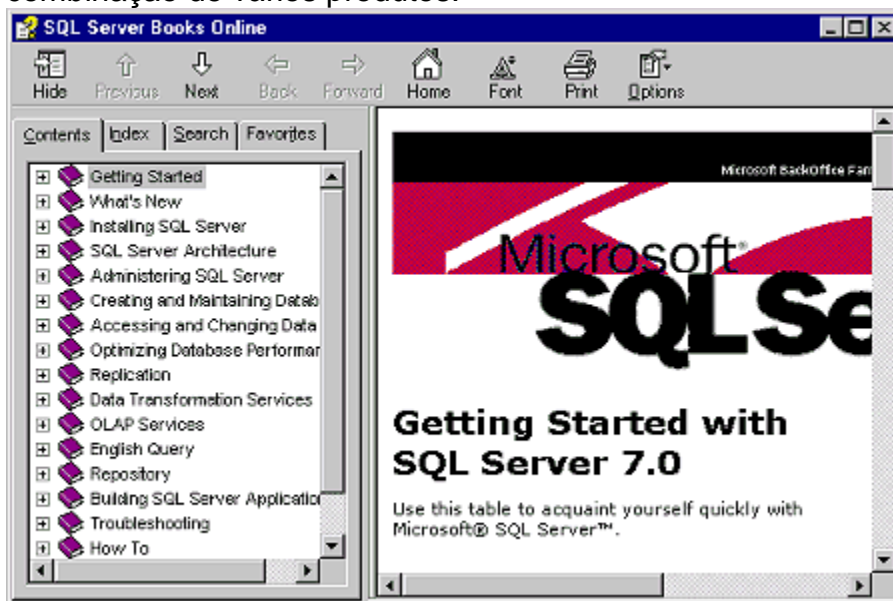
Analyzer (ao lado de Connections), quantas são as conexões estabelecidas. Durante a instalação, se você optou por licenciar o SQL Server como Per Server, seu servidor só suportará tantas conexões simultâneas quantas houverem sido definidas nessa fase da instalação. Então, se você receber uma mensagem como a mostrada abaixo



you must close some connections (query windows) before trying to open a new one, or then, in the menu File | Configure, you must define the maximum number of connections accepted, in the option maximum number of connections (Maximum number of connections).

SQL Server Books Online

The SQL Server Books Online is a great resource to have at hand. The MS books online is becoming the standard way to access help with applications, services and Microsoft programming languages. The online books are a place where you will find everything about a particular product, and in some cases, a combination of several products.



Books Online has a simple navigation, as shown below.

The books online reminds the MMC or the Windows Explorer. The left side has a view of the tree structure while the right side has an IE (Internet Explorer navigator). The pages shown on the right side are simple HTML pages (you can even view the source code).

Search by content [Contents]

A search by content is like searching in a book based on the table of contents. The table of contents is visible as a tree structure. Clicking on a book on the left side of the window causes the book to open, revealing pages, or chapters. Clicking on a page and the page will be displayed on the right side of the window. Clicking on a chapter (which seems like another book) will open more chapters, and more pages on the left side.

Procura por índice [Index]

A procura pelo índice vai pesquisar todas as palavras que foram indexadas quando da criação do material de ajuda. Ocasionalmente, você pode não encontrar o item que você está procurando e precisará de uma procura mais geral. Use a guia Pesquisar [Search] para pesquisas mais gerais.

Usando a guia Pesquisar [Search]

A guia de consulta permite que você digite uma palavra e será pesquisado em todos os documentos atrás daquela palavra. O lado esquerdo da tela vai ser preenchido com os documentos que contém a(s) palavra(s) que você pediu.

Guia Favoritos [Favorites]

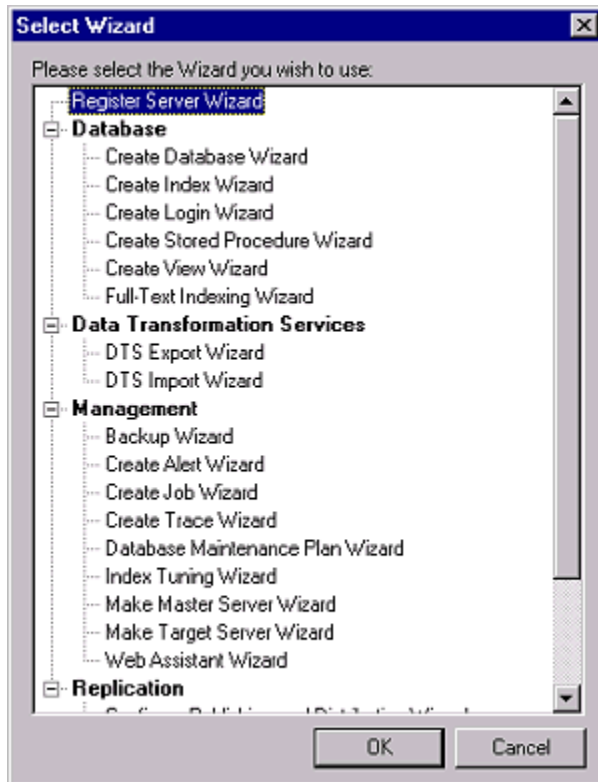
A guia favoritos é um livro de marcadores bem fácil de se usar. Se você gostou da informação que você encontrou e quer torná-las mais fácil de ser encontrada posteriormente, simplesmente selecione a guia Favoritos. A parte inferior da guia vai mostrar-lhe o tópico atual. Clique em Add para colocá-lo como parte do seu marcador. Se você precisar de excluir um item, simplesmente selecione-o e aperte o botão Remove.

Se você está inseguro a respeito de como algumas das ferramentas funcionam, ou apenas esqueceu, lembre-se de que os assistentes são ótimas ferramentas para o aprendizado. Eles fazem um trabalho excelente de apontar as tarefas necessárias que estão sendo realizadas. Depois de executar os assistents algumas vezes, você terá uma boa idéia do processo como um todo. Mesmo que você não tenha, use esta apostila ou os livros on-line.

Assistentes do SQL Server [SQL Server Wizards]

Os assistentes do SQL Server te ajudam a criar seus objetos de banco de dados e serviços sem precisar de estar sempre olhando em suas notas sobre como fazer algo corretamente. Ter um assistente para te encaminhar em cada tarefa é uma maneira ótima de se aprender o que necessita ser feito. Quase tudo tem um assistente. Você já deve alguma vez ter evitado uma certa tarefa só porque você não tinha tempo para pesquisar como realizá-la. Como você deve saber, a não ser que você conheça o procedimento completamente, você cometerá algum erro ao realizá-lo. Seu tempo já é bem escasso sem ter que se recuperar de erros. Se você não estiver certo do que você está fazendo, ou se você está fazendo certo, deixe o assistente ser seu guia. Alguns desses assistentes são simples, enquanto outros são mais complexos. Todos os assistentes estão disponíveis no Enterprise Manager. Para ver uma lista dos assistentes disponíveis, selecione algum servidor, e no menu Ferramentas [Tools] do Enterprise Manager, escolha a opção Assistentes [Wizards]. Aparece a janela mostrada abaixo:

Aqui serão discutidos alguns assistentes e os passos principais para o assistente, dando o máximo de detalhe possível, mas sem exagerar. Bem, assistentes estão aí pra te ajudar, e supõe-se que eles tornem as tarefas difíceis mais fáceis. Então, vamos começar.



Assistente de registro de servidor [Register Server Wizard]

O Assistente de registro de servidor obviamente, registra seu servidor. O ato de registrar o seu servidor é o processo de contar ao SQL Server o nome do servidor que você quer registrar, o tipo de segurança que você está utilizando, seu nome de login e senha (se não estiver usando autenticação do NT), e o grupo de servidor do qual esse servidor vai fazer parte. Você já deve ter utilizado esse assistente em Registrando um servidor.

Você pode criar um novo grupo de servidor quando da criação de um novo banco de dados. Um servidor somente pode existir como parte de um grupo.

Assistente de segurança [Security Wizard]

O assistente de segurança automatiza a tarefa de criação de logins para um servidor SQL Server. O assistente te permite especificar ou uma conta de usuário do NT ou criar uma autenticação do SQL Server. Veja mais sobre segurança.

Assistente de criação de bancos de dados [Create Database Wizard]

O assistente de criação de banco de dados automatiza a tarefa de criar um banco de dados. O assistente te acompanha nos passos principais para criação de um banco de dados. Estes passos incluem o nome do banco de dados, os arquivos do banco de dados, os arquivos de log, o tamanho inicial dos arquivos, e como esses arquivos irão crescer. Tabelas não são criadas aqui. Elas são criadas por comandos Transact-SQL ou clicando com o botão direito em um banco de dados e escolhendo Nova Tabela [New Table]. Saiba mais sobre a criação de bancos de dados e tabelas.

Assistente de criação de alertas [Create Alerta Wizard]

Quando ocorre um erro ou quando o SQL Server dispara um evento, eles são guardados no log de eventos de aplicação, com o nome de SQL Server. O SQL Server Agent lê o log de eventos e faz uma comparação dos eventos com um alerta (que você define). Se houver uma combinação, um alerta é disparado.

Assistente de criação de trabalhos [Create Job Wizard]

Um trabalho é algo que é executado com uma frequência regular. O trabalho pode ser um comando Transact-SQL, um executável, ou mesmo um script (VB-Script). Por exemplo, você pode querer limpar todas as visitas que o seu banco de dados monitora para uma página Web. Isso pode ser feito a cada dia, mês, semana ou ano. O trabalho pode ser simples ou bastante complexo.

Assistente de plano de manutenção de banco de dados [Database Maintenance Plan Wizard]

Este assistente cria uma série de trabalhos que ajudam o seu banco de dados a funcionar melhor. Por exemplo, ele pode agendar a realização de backups em uma base regular e checar por quaisquer inconsistências. Veja um exemplo de seu uso Agendando um backup completo de banco de dados ou de log de transações utilizando um assistente.

Assistente de criação de índices [Create Index Wizard]

Índices são criados automaticamente quando você cria restrições PRIMARY e UNIQUE; entretanto, você também pode querer criar índices em outros campos que serão consultados com frequência. Este assistente permite que você selecione uma tabela e crie índices em um campo dado. Se índices já existem, o assistente te notificará disso. Você pode especificar o índice como um índice clusterizado (se algum ainda não existe), um índice não-clusterizado, índice único, os fatores de preenchimento também podem ser definidos. Se um item não puder ser indexado, tal como um tipo de dados imagem, que armazena as informações em um formato binário (que não é bom para indexação), o SQL Server bloqueará esse campo para indexação. Aprenda mais sobre índices.

Assistente de criação de procedimentos armazenados [Create Stored Procedure Wizard]

Procedimentos armazenados são consultas compiladas. Por que você as quereria e o que elas são? Toda vez que você executa uma consulta, o SQL Engine deve examinar a consulta e certificar-se de que todos os campos e tabelas são válidos. Então, ele deve se decidir a executar a consulta.

Quando você cria um procedimento armazenado, tudo isso é feito de uma vez. O fato de não ter o trabalho adicional de encontrar o melhor caminho a tomar quando da execução de uma consulta pode melhorar o desempenho. Ver mais detalhes em Procedimentos Armazenados.

Assistente de criação de visões [Create View Wizard]

Uma visão é uma tabela virtual que representa uma maneira diferente de se visualizar uma tabela. Ela pode ser usada por vários motivos, incluindo mas não limitada a mostrar apenas a informação que interessa em uma tabela muito grande. Permissões de segurança podem ser administradas para permitir que apenas os pessoal de RH veja toda a tabela que mostra salários, enquanto usuários normais enxergam apenas a informação básica na visão. Veja mais sobre visões.

Assistente de Importação/Exportação DTS [DTS Import/Export wizard]

O assistente DTS [Data Transformation Services - Serviços de Transformação de Dados] permite que você utilize facilmente o DTS para importar ou exportar informações heterogêneas utilizando OLE DB e ODBC; você também pode copiar esquemas e dados de bancos de dados entre bancos de dados relacionais.

Dado heterogêneo é o dado que é armazenado em diferentes formatos de arquivo. A utilização do DTS com OLE DB e ODBC te permite recuperar dados de um formato de arquivo e usá-los com outro formato.

Web Assistant Wizard

Este assistente gera páginas HTML baseado em dados do servidor SQL Server, consultas, procedimentos armazenados, e por aí vai. Os arquivos HTML podem ser publicados de modo que sejam visíveis na intranet da companhia ou na Internet para que o mundo veja. As páginas por si não consultam o banco de dados e assim não são dinâmicas. Entretanto, pode-se criar trabalhos que recriem as páginas regularmente, ou um gatilho poderia lançar um atarefa para recriar a página quando um item fosse inserido, atualizado ou excluído. Veja o uso desse assistente.

4 - Fundamentos de arquitetura do SQL Server

O Catálogo do Sistema

Componentes do Banco de Dados

Estrutura dos bancos de dados

Objetivos:

- Conhecer os conceitos de alocação de espaço usados pelo SQL Server;
- Ter uma visão geral dos itens que compõem o catálogo do sistema;
- Saber o que é um banco de dados e o que ele contém.

O Catálogo do Sistema

Um *banco de dados* é uma coleção de tabelas e outros objetos relacionados. Existem dois tipos de banco de dados: os bancos de dados *do sistema* são usados pelo SQL Server para operar e gerenciar o sistema e os bancos de dados *do usuário* são usados para armazenar os seus próprios dados. O catálogo do sistema é composto de tabelas no banco de dados *master*.

Bancos de Dados do Sistema

Ao instalar o SQL Server, são criados os seguintes bancos de dados do sistema: *master* Controla os bancos de dados do usuário e a operação do SQL Server. Tem como tamanho inicial 16 MB. É importante manter um backup atualizado desse banco de dados. Contém informações sobre:

- Contas de login
- Processos em execução
- Mensagens de erro
- Bancos de dados criados no servidor
- Espaço alocado para cada banco de dados
- Travas [locks] de linha ativas
- Espaço alocado para cada banco de dados
- Procedimentos armazenados do sistema

model É um modelo usado para criação de novos bancos de dados, que pode ser usado para definir padrões, como autorizações default de usuário, opções de configuração, tipos de dados etc. Sempre que um banco de dados do usuário é criado, o conteúdo de *model* é copiado para ele. Seu tamanho inicial é 2.5 MB. Esse modelo pode ser alterado.

tempdb Usado para armazenar tabelas temporárias e resultados intermediários de consultas. Geralmente o seu conteúdo é excluído sempre que um usuário se

desconecta. Ele cresce automaticamente conforme é necessário. Seu tamanho inicial é de 8 Mb.

msdb Usado pelo serviço SQLServerAgent, para controlar tarefas como replicação, agendamento de tarefas, backups e alertas. Contém algumas tabelas de sistema, que armazenam informações usadas pelo SQLExecutive. Seu tamanho inicial é 12 Mb.

Tabelas do Sistema

As tabelas do sistema, armazenadas no banco de dados *master* e em cada banco de dados de usuário, contêm informações sobre o SQL Server e sobre cada banco de dados de usuário. Existem 17 tabelas em cada banco de dados que formam o *catálogo do banco de dados*. Todas começam com o prefixo *sys* e contêm as seguintes informações:

<i>syscolumns</i>	Informação sobre cada coluna de cada tabela, e cada parâmetro de procedimento.
<i>syscomments</i>	Para cada objeto de banco de dados (visão, regra, default, trigger, procedimento) contém o texto de sua definição.
<i>sysconstraints</i>	Inclui informações sobre todas as restrições usadas no banco de dados.
<i>sysdepends</i>	Registra as dependências entre objetos do banco de dados.
<i>sysfilegroups</i>	Tem uma linha para cada grupo de arquivos armazenado em um banco de dados.
<i>sysfiles</i>	Informações sobre cada arquivo de um banco de dados.
<i>sysforeignkeys</i>	Informações sobre todas as restrições de chaves estrangeiras encontradas em todas as tabelas de um banco de dados.
<i>sysfulltextcatalogs</i>	Lista todos os catálogos de texto completo para esse banco de dados.
<i>sysindexes</i>	Informação para cada índice criado e para cada tabela sem índices, além de informações para cada tabela que possui colunas <i>text</i> ou <i>image</i> .
<i>sysindexkeys</i>	Informação sobre as chaves e as colunas de um índice.
<i>sysmembers</i>	Informações sobre os membros de cada papel.
<i>sysobjects</i>	Informação sobre cada objeto do banco de dados (tabelas, visões, procedimentos, regras, defaults e gatilhos).
<i>syspermissions</i>	Informação sobre permissões atribuídas a usuários, grupos e papéis em um banco de dados.
<i>sysprotects</i>	Permissões atribuídas às contas de segurança.
<i>sysreferences</i>	Informação sobre toda restrição de integridade referencial usada numa coluna ou tabela de um banco de dados.
<i>systypes</i>	Informação sobre cada tipo de dados (do sistema ou definido pelo usuário).
<i>sysusers</i>	Informação sobre cada usuário que pode ter acesso ao banco de dados.

Existem também tabelas localizadas apenas no banco de dados *master*, que compõem o *catálogo do sistema*. Elas contêm as seguintes informações:

sysallocations Informações sobre cada unidade de alocação gerenciada pelo

	SQL Server
<i>sysaltfiles</i>	Informações sobre cada arquivo gerenciado pelo SQL Server
<i>syscharsets</i>	Informação sobre conjuntos de caracteres [character sets] e ordens de classificação [sort orders].
<i>sysconfigures</i> , <i>syscurconfigs</i>	Parâmetros de configuração do SQL Server.
<i>sysdatabases</i>	Informação sobre os bancos de dados existentes.
<i>sysdevices</i>	Informação sobre os dispositivos, tais como o dispositivo de fita.
<i>syslanguages</i>	Idiomas suportados pelo servidor.
<i>syslockinfo</i>	Travas (locks) ativas.
<i>syslogins</i>	Contas de login.
<i>sysmessages</i>	Mensagens de erro do sistema
<i>sysoledbusers</i>	Contém uma linha para cada usuário e senha mapeados em um servidor.
<i>sysperfinfo</i>	Informação sobre os monitores de performance.
<i>sysprocesses</i>	Processos em execução
<i>sysremotelogins</i>	Contas de login remotas.
<i>sysservers</i>	Servidores remotos conhecidos.

Procedimentos Armazenados do Sistema

Um *procedimento armazenado* [stored procedure] é uma seqüência de comandos da linguagem Transact-SQL, compilados e armazenados num banco de dados. Os *procedimentos armazenados do sistema* [system stored procedures] são fornecidos pelo SQL Server, armazenados no banco de dados *master* e automatizam várias tarefas comuns de gerenciamento.

Por exemplo, o procedimento *sp_databases* mostra quais os nomes de bancos de dados existentes. Para executar esse procedimento, use o Query Analyzer (Iniciar | Programas | Microsoft SQL Server 7.0 | Query Analyzer). Na página "Query" digite:

sp_databases
E clique no botão Execute. Ele mostra um resultado como:

DATABASE_NAME	DATABASE_SIZE	REMARKS
-----	-----	-----
master	17408	(null)
model	1024	(null)
msdb	8192	(null)
pubs	3072	(null)
tempdb	2048	(null)

(5 row(s) affected)

Outro procedimento útil é *sp_helpdb*. Ele mostra informações sobre um banco de dados.

Sintaxe:

sp_helpdb [Banco de Dados]

Exemplo:

sp_helpdb master

O resultado será algo como:

name	db_size	owner	dbid	created	status
------	---------	-------	------	---------	--------

-----	-----	-----	-----	-----	-----	-----	-----
master		17.00 MB	sa	1	Apr 3 1996	trunc. log on chkpt.	
name	file	filename	filegrou	size	maxsize	grow	usage
-----	id	-----	p	----	-----	th	-----
-----	---	-----	-----	9024	Unlimit	---	data
master	1	F:\MSSQL7\data\master.mdf	PRIMARY	KB	ed	10%	only
mastlog	2	F:\MSSQL7\data\mastlog.ldf	NULL	1280	Unlimit	10%	log
				KB	ed		only

Note que todos os procedimentos armazenados do sistema têm nomes que começam com 'sp_' (abrev. de system procedure). Ao executar um procedimento que inicie com 'sp_' o procedimento será procurado no banco de dados atual, se não for encontrado ele será procurado no banco de dados Master.

Quando um procedimento inicia com 'xp_' ele é um procedimento estendido, quer dizer que não foi escrito em SQL, mas foi compilado como parte de uma DLL.

Por exemplo, o procedimento xp_cmdshell executa um programa. Para executar o procedimento é necessário que esteja posicionado no banco de dados Master.

Sintaxe

xp_cmdshell 'nome_arquivo'

Onde:

nome_arquivo é o nome do programa que deseja executar.

Exemplo:

xp_cmdshell 'calc.exe'

Para mais informações consulte SQL Server Books Online em Transact-SQL item Stored Procedures ou no Help existe tópicos para cada procedimento.

Componentes do Banco de Dados

Um banco de dados é composto de *objetos*, *índices*, *tipos de dados* e *restrições*: Cada objeto tem uma linha correspondente na tabela sysobjects. Seu tamanho mínimo é 1 Mb.

Objetos

Um objeto contém dados ou interage com os dados. Cada objeto tem uma linha correspondente a tabela sysobjects. Existem os seguintes tipos de objetos:

Tabela [table]	Conjunto de linhas, compostas de colunas. Cada coluna armazena um item de dado.
Visão [view]	Uma forma alternativa de visualizar dados em uma tabela ou mais.
Default	Um valor que é inserido numa coluna caso não tenha sido informado um valor.
Regra [rule]	Valida os dados que podem ser inseridos em uma coluna.
Procedimento armazenado [stored procedure]	Uma seqüência de comandos SQL, compilados e armazenados no banco de dados.
Gatilho [trigger]	Uma seqüência de comandos executados automaticamente

quando os dados são modificados numa tabela.

Índices

Um índice é composto de ponteiros para os dados, ordenados pelo valor de uma ou mais colunas. Através de um índice, é possível acessar mais rapidamente os dados, dado o valor de algumas colunas.

Tipos de dados

Um tipo de dados especifica quais os valores que podem ser armazenados em uma coluna.

Restrições [constraints]

Uma restrição reforça a integridade dos dados em uma tabela, ou entre duas tabelas, controlando quais dados podem ser inseridos.

Estrutura dos bancos de dados

Todo banco de dados do SQL Server é constituído de dois ou mais arquivos físicos de sistema operacional. Podem haver três tipos diferentes de arquivos físicos:

- Arquivos primários: Todo banco de dados inclui ao menos um arquivo primário, que é feito para armazenar todos os objetos de banco de dados, tais como tabelas e índices. Este arquivo também é usado para apontar para o resto dos arquivos que constituem o banco de dados.
- Arquivo secundário: Um banco de dados só terá um arquivo secundário se o arquivo primário não for grande o suficiente para armazenar todos os dados. Um banco de dados pode ter um, ou muitos arquivos secundários.
- Arquivo de log: Todo banco de dados tem um arquivo de log, que é usado para registrar todas as informações antes que elas sejam escritas em um arquivo primário ou secundário. Esses dados são utilizados para ajudar na recuperação, no caso de um problema com o banco de dados. Um banco de dados pode ter um ou vários arquivos de log se o arquivo de log original ficar sem espaço.

Nomes lógicos e físicos

Um dispositivo de banco de dados é um arquivo do sistema operacional, por exemplo, o banco de dados MASTER é o arquivo C:\MSSQL7\DATA\master.mdf, localizado no servidor. Como default, os bancos de dados são criados no diretório C:\MSSQL7\DATA, mas podem ser criados em qualquer drive ou diretório acessível ao SQL Server.

Cada banco de dados tem um *nome físico* (o caminho e nome do arquivo) e um *nome lógico* (nome usado dentro do SQL Server). Os dois não precisam estar relacionados. Por exemplo, ao criar um banco de dados chamado VENDAS, o SQL Server, por default, cria um arquivo chamado VENDAS.mdf, no diretório C:\MSSQL7\DATA. O nome físico pode ser qualquer nome suportado pelo sistema operacional. O nome lógico pode ter até 30 caracteres e geralmente só contém letras e números (espaços não são recomendados).

Subdivisões de espaço

Cada banco de dados é dividido ainda em:

- Unidades de alocação:** Cada unidade de alocação tem 512 Kb (meio megabyte). Um banco de dados ocupa sempre um número inteiro de unidades de alocação.
- Extents:** Um *extent* tem 64 Kb. Cada objeto de banco de dados (tabelas, índices) ocupa um número inteiro de *extents*
- Página [page]:** Uma página (8Kb) é a unidade mais básica de armazenamento. Um objeto do banco de dados sempre cresce em páginas e em alguns casos pode ficar fragmentado, disperso em páginas distantes uma da outra.

Arquivos predefinidos

Ao instalar o SQL Server, são criados quatro bancos de dados, com os seguintes nomes lógicos:

MASTER: composto pelos arquivos *master.mdf* e *mastlog.ldf* (arquivo primário e de log)

MSDB: composto pelos arquivos *msdbdata.mdf* e *mastlog.ldf* (arquivo primário e de log, que por sinal é o mesmo arquivo de log do banco de dados *master*)

MODEL: composto pelos arquivos *model.mdf* e *modellog.ldf*.

TEMPDB: composto pelos arquivos *tempdb.mdf* e *templog.ldf*

5 - Criando Bancos de Dados

Gerenciando Bancos de Dados

Arquivos e grupos de arquivos

Criando Tabelas

Alterando Estrutura das Tabelas

Definindo opções de bancos de dados

Considerações para melhor gerenciamento

Documentando a criação de bancos de dados

Objetivos:

- Aprender a gerenciar bancos de dados, criando, alterando ou excluindo-os com o Enterprise Manager ou com comandos SQL;
- Aprender a criar tabelas e alterar sua estrutura.

Gerenciando Bancos de Dados

Você cria um banco de dados [database] definindo o seu nome, nome do arquivo, tamanho inicial, tamanho máximo e taxa de crescimento. Inicialmente apenas o administrador do sistema (SA) pode criar, modificar o tamanho e excluir os bancos de dados, mas ele pode conceder permissões a outros usuários para isso. Em versões anteriores do SQL Server, havia a necessidade de criação de dispositivos [devices], e dentro destes deviam ser criados os bancos de dados. Bem, não existem mais dispositivos no SQL Server 7.0.

Depois de fazer excluir ou fazer alterações no do banco de dados é recomendável fazer backup do banco de dados **master**, porque ele contém informações sobre cada banco de dados. Mais especificamente, a tabela de sistema *sysdatabases*, do banco de dados master, armazena as informações sobre todos os bancos de dados. Mais adiante, discutiremos com mais detalhes as tabelas de sistema.

O Log de Transações [Transaction Log]

Quando você cria um banco de dados, é criado também um *log de transações* [transaction log] para esse banco de dados. Esta é uma área reservada onde todas as alterações feitas no banco de dados são registradas. Qualquer comando SQL que modifica os dados registra as alterações *antes* no log de transações, *depois* nas tabelas alteradas.

Quando é executado um comando que altera os dados (insere, altera ou exclui linhas numa tabela), essa alteração é salva primeiro no log de transações, escrevendo diretamente em disco. Os dados são alterados apenas em memória.

Periodicamente, o SQL Server faz um *checkpoint*, um processo que grava em disco as alterações feitas em memória. (Geralmente um checkpoint é feito uma vez por minuto). O log de transações permite recuperar o banco de dados a um estado consistente, em caso de uma pane no sistema. Sempre que o SQL Server inicia, ele verifica o log de transações para saber se alguma alteração foi iniciada, mas não salva nos dados. As transações que não foram confirmadas (*committed*) são canceladas.

Pode ser interessante colocar o log fisicamente em um disco diferente dos dados, o que melhora o desempenho, pois as operações de E/S podem ser feitas simultaneamente nos dois.

Por padrão, ao se criar um banco de dados, o log de transações é criado com 25% do tamanho do banco de dados. Pode-se mudar o tamanho do mesmo. Recomenda-se alocar para o log de 10 a 25% do tamanho do banco de dados.

Criando bancos de dados com o Enterprise Manager

Vamos criar um banco de dados com o SQL Enterprise Manager.

No SQL Enterprise Manager, conecte-se ao servidor desejado. Clique em "Databases" com o botão direito e selecione **New Database...**

Em "Name" coloque Exemplo. Note que ao definir o nome, o nome do arquivo [File Name] (que não é necessariamente o mesmo nome definido em "Name", apesar de poder ser), muda para o nome que você está digitando seguido de *_data*. Se você clicar na guia Transaction Logs, verá que o arquivo de log está sendo criado, com o nome do banco de dados seguido de *_log*, e no caminho definido para o banco de dados.

Em "Location", o local no disco onde você quer armazená-lo. Por padrão, o SQL Server define o local como sendo a subpasta \DATA, na pasta de instalação do SQL Server. O tamanho usado, "Size", aparece como 1 Mb por default, substitua esse valor por 10 Mb. O tamanho alocado para o log, por default, é também 1 Mb, como você pode ver clicando na guia Transaction Logs. Mude o tamanho para 2Mb.

A opção "Automatically grow file" determina se o arquivo poderá ser expandido à medida que for ficando cheio. Se essa opção estiver marcada, você pode determinar a taxa de crescimento (a cada vez que ele for expandido, será expandido em N% ou N megabytes) do arquivo [File growth] em porcentagem ou megabytes. Também é possível determinar um tamanho limite para o arquivo [Restrict file growth] ou deixá-lo crescer indeterminadamente [Unrestricted file growth].

Por padrão, como se percebe, o banco de dados tem como propriedades crescer automaticamente, em incrementos de 10 por cento, e sem limite de crescimento.

Clique em **Ok** e aguarde alguns instantes: o banco de dados será criado, com 10 Mb para os dados e 2 Mb para o log de transações.

Nota: O arquivo de banco de dados é gravado com a extensão **.mdf** ou **.ndf**, dependendo se ele for um arquivo primário ou não-primário no banco de dados. Veremos isso melhor em Grupos de Arquivos. O arquivo de log é gravado com a extensão **.ldf**.

Nota: Ao ser criado, um banco de dados é uma cópia do banco de dados **model**. Quaisquer opções ou configurações do banco de dados **model** são copiadas no novo banco de dados.

Criando bancos de dados com comandos SQL

Para gerenciar os bancos de dados com comandos SQL é necessário que se esteja posicionado no banco de dados **master**.

Você também pode criar um banco de dados com o comando SQL, CREATE DATABASE.

Sintaxe

```
CREATE DATABASE nome_bancodedados
[ON {
[PRIMARY] (NAME = nome_lógico_arquivo,
    FILENAME = 'caminho_e_nome_arquivo'
    [, SIZE = tamanho]
    [, MAXSIZE = tamanho_máximo]
    [, FILEGROWTH = taxa_crescimento]
    }[,...n]
]

[LOG ON
    {
    (NAME = nome_lógico_arquivo,
    FILENAME = 'caminho_e_nome_arquivo'
    [, SIZE = tamanho])
    }[,...n]
]
```

Onde:

nome_bancodedados é o nome do banco de dados que se deseja criar.

nome_logico_arquivo é um nome usado para referenciar o arquivo em quaisquer comandos SQL executados depois que o banco de dados tiver sido criado.

PRIMARY: Esta opção especifica o grupo de arquivos primário. O grupo de arquivos primário deve conter todas as tabelas de sistema para o banco de dados. Um banco de dados só pode ter um grupo de arquivo PRIMARY. Se não for especificado algum, o primeiro listado será o primário. (Veremos grupos de arquivos em breve).

FILENAME: Aqui deve-se especificar o caminho e nome do arquivo que você está criando. O arquivo deve estar localizado na mesma máquina que o servidor SQL Server. Ele pode estar em uma unidade de disco diferente contanto que esteja na mesma máquina.

SIZE: Especifica o tamanho em megabytes que você quer alocar para o seu banco de dados. O valor mínimo é 1MB, e o padrão é 3MB para arquivos de dados, e 1MB para arquivos de log. (Obs.: o padrão aqui diz respeito à criação do banco de dados por comandos SQL. Como vimos, quando criado pelo Enterprise Manager, o padrão é 1MB de tamanho tanto para arquivos de banco de dados quanto para arquivos de log).

MAXSIZE: Esta opção lhe permite especificar o tamanho máximo até o qual seu arquivo pode crescer. O padrão permite que seu arquivo cresça até que o disco esteja cheio.

FILEGROWTH: Especifica a taxa de crescimento do arquivo. Este ajuste não pode exceder a configuração de MAXSIZE. Um valor de 0 indica que não é permitido crescimento. O padrão é 10 por cento, significando que a cada vez que o arquivo cresce, será alocado um espaço adicional de 10 por cento para ele. Um banco de

dados que esteja em mais de um arquivo (veja Arquivos para maiores detalhes), só é expandido depois que o último arquivo estiver cheio.

Na opção LOG ON se aplicam as mesmas definições acima, exceto pelo fato de não ser o arquivo de dados, mas sim o arquivo de log de transações que estará sendo criado.

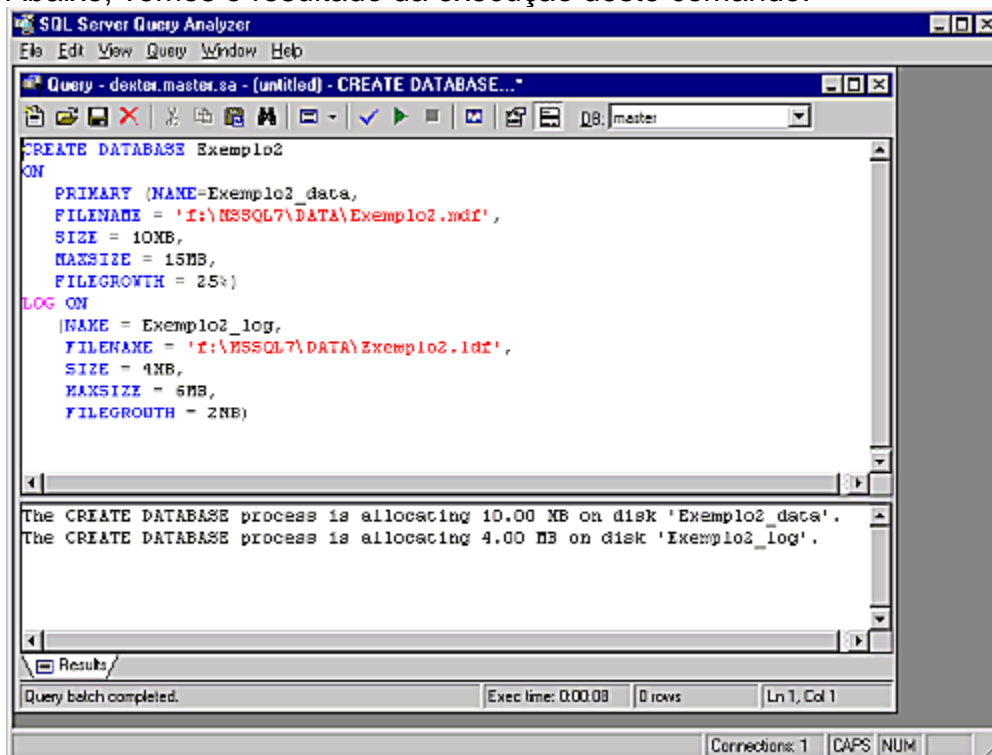
Caso LOG ON seja omitido, é criado um único arquivo de log com um nome gerado pelo sistema e um tamanho que seja 25 por cento da soma dos tamanhos de todos os arquivos de dados para o banco de dados.

Nota: A opção FOR LOAD pode ser adicionada antes da declaração LOG ON. Não recomenda-se usá-la, já que existe apenas para compatibilidade com versões anteriores. Esta opção coloca a opção **for dbo use only** como verdadeira, e o status como para carregar [*for load*]. Essa opção não é necessária, pois no SQL Server 7, temos o comando RESTORE pode recriar um banco de dados como parte de uma operação de recuperação.

Como exemplo, vamos criar um banco de dados, entrando com o seguinte código SQL no Query Analyzer.

```
CREATE DATABASE Exemplo2
ON
    PRIMARY (NAME=Exemplo2_data,
        FILENAME = 'C:\MSSQL7\DATA\Exemplo2.mdf',
        SIZE = 10MB,
        MAXSIZE = 15MB,
        FILEGROWTH = 25%)
LOG ON
    (NAME = Exemplo2_log,
        FILENAME = 'C:\MSSQL7\DATA\Exemplo2.ldf',
        SIZE = 4MB,
        MAXSIZE = 6MB,
        FILEGROWTH = 2MB)
```

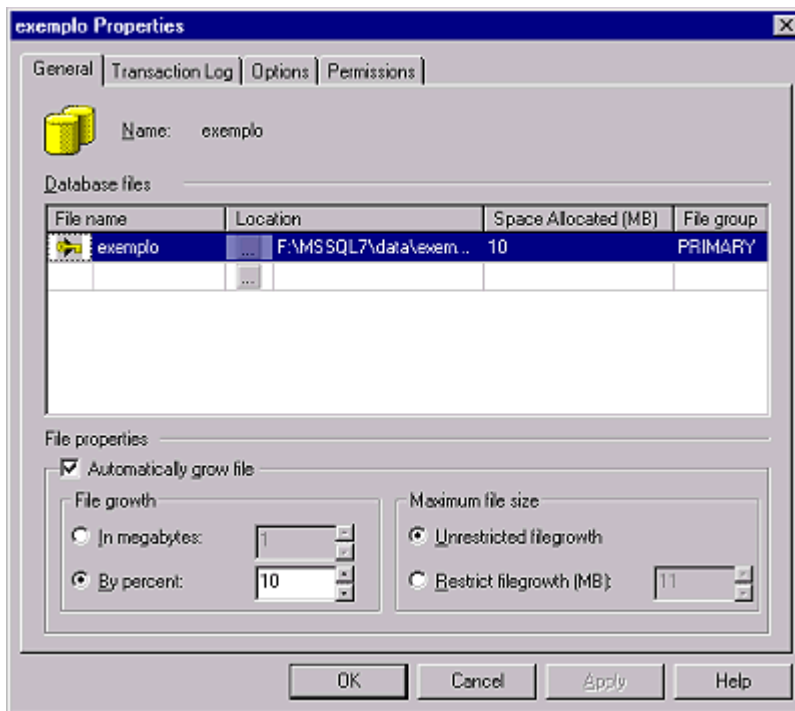
Abaixo, vemos o resultado da execução deste comando:



Criando um banco de dados com o assistente de criação de banco de dados

O assistente de criação de banco de dados te encaminha através dos passos necessários para a criação de um banco de dados e de um log. Para iniciá-lo, faça o seguinte:

1. A partir do Enterprise Manager, com algum servidor selecionado, selecione Tools | Wizards. Todos os objetos que têm assistentes são listados em ordem alfabética.
2. Depois que você selecionar Databases (bancos de dados) a árvore deve se expandir para lhe mostrar uma lista dos assistentes disponíveis para bancos de dados.
3. Selecione Create Database Wizard. Aparece a tela de boas-vindas. Clique em Next para continuar.
4. Escolha o nome do banco de dados, e o local onde ficarão seus arquivos de dados e de log. Clique em Next.
5. Mude o nome do arquivo e seu tamanho inicial, conforme desejar.
6. Por fim aparecem as opções, que são não permitir o crescimento do banco de dados [Do not automatically grow the database]. Caso você escolha por crescer automaticamente o banco de dados [Automatically grow database files], você tem as opções de incrementá-lo em megabytes [Grow the files in Megabytes], ou porcentagem [Grow the files by percent], e definir as taxas de incremento. Por fim, você pode limitar o tamanho máximo do arquivo [Restrict file growth to (MB)] ou deixá-lo crescer o quanto for necessário [Unrestricted File Growth]. Faça suas opções e clique em Next para continuar.
7. Então, defina o nome do arquivo e tamanho inicial para o arquivo de log. Clique em Next para continuar.
8. Finalmente, defina o mesmo citado no item 6 acima, para o arquivo de log. Clique em Next.
9. Aparece a tela final, mostrando-lhe o que você definiu. Se estiver satisfeito, clique em Finish para criar o banco de dados. Caso contrário, clique em Back e altere as opções que achar necessário.



Alterando um banco de dados pelo Enterprise Manager

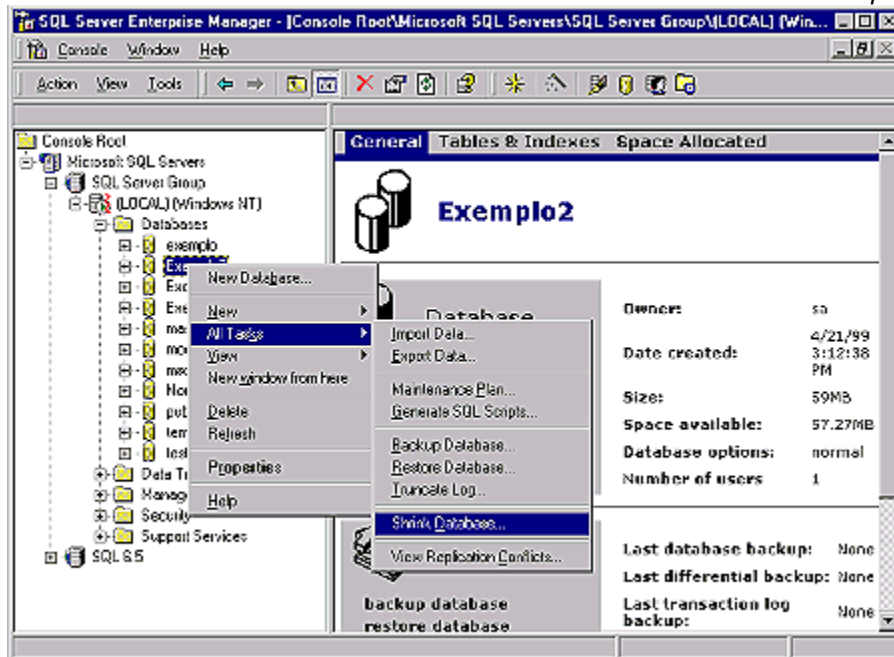
Após criar um banco de dados, você pode aumentá-lo ou reduzi-lo. No Enterprise Manager, clique no banco de dados com o botão direito e em **Properties**. Você verá os tamanhos atuais dos dados e do log e quanto espaço disponível existe em cada um.

Para aumentar o tamanho do banco de dados, informe o novo tamanho em "Space Allocated" (que deve ser maior que o tamanho original). Note que você só pode alterar, na guia General, o espaço alocado para o banco de dados, e as opções de crescimento [File Properties]. É só clicar em Ok ou em Apply que as mudanças são aplicadas.

Vamos aumentar o tamanho do banco de dados de 10 para 11 Mb. Basta para isso, digitar 11 em "Space Allocated", e clicar em Ok.

Para reduzir um banco de dados pelo Enterprise Manager, selecione o banco de dados que se quer reduzir, clique com o botão direito nele, selecione All Tasks | Shrink database, como mostrado abaixo

Na janela que aparece a seguir, você tem algumas opções, que são:



- Reorganize database: esta opção, se selecionada, faz uma espécie de desfragmentação do banco de dados, agrupando as páginas preenchidas, e deixando as páginas livres no fim do arquivo.
- Shrink database files: esta opção disponibiliza o espaço livre que houver no fim do arquivo, para o sistema operacional.

Alterando um banco de dados com comandos SQL

Para expandir o banco de dados, pode-se usar ALTER DATABASE. Para utilizar este comando, deve-se estar posicionado no banco de dados **master**:

Sintaxe

```
ALTER DATABASE nome_bancodedados
MODIFY FILE
(NAME = nome_logico_arquivo,
    SIZE = novo_tamanho
)
```

Onde:

nome_bancodedados é o nome do banco de dados que se deseja alterar.

nome_logico_arquivo é o nome lógico dado ao arquivo na sua criação.

novo_tamanho é o tamanho novo do banco de dados. Esse tamanho é fornecido em MegaBytes.

Por exemplo:

Vamos fazer o banco de dados aumentar para 12 Mb. Para isso, digite:

```
ALTER DATABASE Exemplo2
MODIFY FILE
(NAME = Exemplo2_data,
    SIZE = 12MB
```

)
 Para expandir o log de transações do banco de dados Exemplo2 para 3MB, faça:

```
ALTER DATABASE Exemplo2
MODIFY FILE
  (NAME = Exemplo2_log,
   SIZE = 3MB
  )
```

Com o comando SQL ALTER DATABASE, também é possível acrescentar arquivos ao banco de dados. Para isso, usa-se a opção ADD FILE. Vamos adicionar um arquivo secundário ao banco de dados Exemplo2, com 4MB iniciais, e tamanho máximo de 7MB. Veja mais sobre arquivos secundários.

```
ALTER DATABASE Exemplo2
  ADD FILE (NAME='Exemplo2_dados2',
  FILENAME='C:\mssql7\data\Exemplo2.ndf',
  SIZE=4MB,
  MAXSIZE=7MB)
```

Para reduzir o tamanho de um banco de dados, usa-se o comando DBCC SHRINKDATABASE. Este comando reduz o tamanho de todos os arquivos de dados no banco de dados mas não diminui o tamanho dos arquivos de log.

Sintaxe

```
DBCC SHRINKDATABASE
  (Nome_BancodeDados ,porcentagem_final
  [, {NOTRUNCATE | TRUNCATEONLY}])
  )
```

Onde:

nomebancodedados é o nome do banco de dados que se deseja alterar o tamanho.

porcentagem_final é a porcentagem de espaço livre a ser deixada no banco de dados depois que o mesmo for reduzido.

NOTRUNCATE faz com que o espaço liberado permaneça nos arquivos de banco de dados. Se não especificado, o espaço liberado é tornado disponível para o sistema operacional.

TRUNCATEONLY faz com que o espaço não utilizado em arquivos de dados seja liberado para o sistema operacional e encolhe o arquivo até o último espaço utilizado, reduzindo o tamanho do arquivo sem mover quaisquer dados. Nenhuma tentativa é feita para relocar colunas em páginas não-alocadas. *porcentagem_final* é ignorado quando TRUNCATEONLY for utilizado.

Não se pode diminuir o tamanho do banco de dados para um tamanho menor do que o tamanho mínimo do arquivo, que é especificado quando o arquivo foi originalmente criado.

Nota: O banco de dados não pode ficar menor que o tamanho do banco de dados **model**.

Pode-se também reduzir todo o banco de dados usando o comando DBCC SHRINKFILE. Este comando reduz o tamanho de um arquivo de dados específico em um banco de dados.

Sintaxe

```
DBCC SHRINKFILE ({nome_arquivo|id_arquivo}, [novo_tamanho][,
  {EMPTYFILE | NOTRUNCATE | TRUNCATEONLY}])
```

Todas as opções já foram explicadas para o comando DBCC SHRINKDATABASE, com duas exceções:

- Aqui se especifica o nome do arquivo, e não do banco de dados.
- O novo tamanho é dado em MB, não em porcentagem como acima.
- A opção EMPTYFILE migra todos os dados do arquivo especificado para outros arquivos no mesmo grupo de arquivos. O SQL Server não permite mais que dados sejam colocados no arquivo em que foi utilizada a opção EMPTYFILE. Esta opção permite que o arquivo seja excluído com o comando ALTER DATABASE, com a opção REMOVE FILE.

Nota: Quando se usa qualquer das opções EMPTYFILE, NOTRUNCATE ou TRUNCATEONLY, não se especifica o novo tamanho.

Como exemplo, vamos reduzir o arquivo secundário do banco de dados Exemplo2 (Exemplo2.ndf), que chamamos de Exemplo2_Dados2 para 2MB. Para isso, entre com o seguinte comando no Query Analyzer:

```
DBCC SHRINKFILE (Exemplo2_Dados2,2)
```

O banco de dados a ser reduzido não precisa estar em modo de único usuário, outros usuários podem estar trabalhando com o banco de dados quando ele é reduzido.

Excluindo um banco de dados pelo Enterprise Manager

Para excluir um banco de dados no Enterprise Manager, clique no Banco de Dados e com o botão direito, clique em **Delete**. Confirme a exclusão. Após excluir um banco de dados, não é possível recuperar os dados, a não ser que você tenha feito um backup. Como iremos usar este banco de dados nos exemplos posteriores, crie-o novamente com as mesmas características que foi criado anteriormente.

Excluindo um banco de dados com comandos SQL

Para excluir um banco de dados, usa-se:

Sintaxe

```
DROP DATABASE nome_bancodedados[, nome_bancodedados...]
```

Onde:

nome_bancodedados é o nome do banco de dados que se deseja excluir.

Você pode excluir um banco de dados ou múltiplos banco de dados.

Exemplo:

Para excluir vários banco de dados ao mesmo tempo, usa-se:

```
DROP DATABASE Exemplo, Exemplo2
```

Se quiséssemos ter excluído somente o banco de dados Exemplo2, teríamos usado:

```
DROP DATABASE exemplo3
```

Se executar este comando crie o banco de dados Exemplo novamente, pois iremos usá-los nos exemplos posteriores.

Arquivos e grupos de arquivos

Arquivos e grupos de arquivos são a nova estrutura de armazenamento do SQL Server. Um banco de dados é armazenado em uma estrutura de arquivos, e então um grupo de arquivos padrão é criado quando você cria seu banco de dados. Outros arquivos podem ser acrescentados ao seu projeto. Estes arquivos podem ser agrupados com base em grupos de arquivos definidos pelo usuário. Na maioria dos casos, você usará apenas o grupo de arquivos padrão, e ter apenas um arquivo para a sua estrutura de banco de dados. Como veremos, entretanto, assim que você estiver familiarizado com o conceito de arquivo e grupo de arquivo, o desempenho pode ser aumentado e a administração tornada mais fácil.

O uso de grupos de arquivos é uma técnica avançada de projeto de banco de dados. Você deve compreender a estrutura, transações, consultas, e dados de seu banco de dados profundamente de modo a determinar a melhor maneira de armazenar tabelas e índices em grupos de arquivos específicos. Em muitos casos, o uso das capacidades de sistemas RAID (Redundant Array of Inexpensive Disks) fornece quase o mesmo ganho em desempenho que você poderá obter com o uso de grupos de arquivos sem o encargo administrativo extra de definir e gerenciar grupos de arquivos.

Arquivos

Como dito anteriormente, o SQL Server cria bancos de dados e logs baseado em uma estrutura de arquivos ao invés da especificação de Dispositivo [Device] de versões anteriores. Isso permite que os arquivos de bancos de dados e de log sejam escalados com maior facilidade. Cada arquivo pode ser usado por apenas um banco de dados. Ele não pode ser compartilhado entre vários bancos de dados. Quando um banco de dados for excluído, seja através de DROP DATABASE ou do Enterprise Manager, o arquivo associado também é excluído.

O SQL Server tem três tipos de arquivos:

- **Primário.** O arquivo primário é o ponto de partida do banco de dados e aponta para o resto dos arquivos no banco de dados. Todo banco de dados deve ter ao menos um arquivo de dados primário. A extensão padrão e recomendada é **.mdf**.
- **Secundário.** Arquivos de dados secundários são os outros (não-primários) arquivos no banco de dados. Alguns bancos de dados podem não ter quaisquer arquivos secundários de dados, enquanto outros podem ter múltiplos arquivos secundários. A extensão padrão e recomendada é **.ndf**.

- **Log.** Um arquivo de log é uma área de armazenamento para todas as mudanças nos bancos de dados. Tdo banco de dados deve ter pelo menos um arquivo de log. A extensão recomendada é **.ldf**.

Nota: o SQL Server não te força a usar a extensão recomendada para os arquivos. Elas existem simplesmente para te ajudar a gerenciar seus arquivos de bancos de dados.

Grupos de arquivos

Os arquivos de bancos de dados são agrupados para fins de alocação e administração. Alguns sistemas podem ter seu desempenho aumentado pelo controle do armazenamento de dados e índices em unidades e disco específicas. Grupos de arquivos podem auxiliar nesse processo. O administrador de sistema pode criar grupos de arquivos para cada unidade de disco, e então definir que certas tabelas, índices, entre outros objetos, sejam armazenados em grupos de arquivos específicos. Quando um banco de dados é criado, o grupo de arquivos primário contém o arquivo primário.

Outros grupos de arquivos (grupos de arquivos definidos pelo usuário) podem ser criados e agrupados para fins de alocação e administração. Você cria um grupo de arquivos como uma coleção nomeada de arquivos. Nenhum arquivo pode ser membro de mais de um grupo de arquivos.

Existem três tipos de grupos de arquivos

- **Primário.** O grupo de arquivo primário contém o arquivo de dados primário e quaisquer outros arquivos de dados não atribuídos a outros grupos de arquivos. Todas as tabelas de sistema são armazenadas no grupo de arquivos primário.
- **Definido pelo usuário [User-defined].** Grupos de arquivos definidos pelo usuário são criados usando a palavra-chave FILEGROUP quando se utiliza os comandos CREATE DATABASE ou ALTER DATABASE.
- **Padrão.** O grupo de arquivos padrão contém todas as tabelas e índices que não têm um grupo de arquivo especificado quando eles são criados. Em cada banco de dados, apenas um grupo de arquivos de cada vez pode ser o grupo de arquivos padrão.

Nota: Arquivos de log nunca fazem parte de um grupo de arquivos. O espaço do log é gerenciado separadamente do espaço dos dados.

Vamos agora criar um grupo de arquivos chamado Grupo1_Exemplo2 no banco de dados Exemplo2, e então adicionar um arquivo secundário chamado Teste ao banco de dados Exemplo2, com o nome físico de TesteGrupo1_Exemplo2.ndf, com 3MB de tamanho, mas este arquivo será adicionado ao grupo Grupo1_Exemplo2. Entre com o seguinte código SQL:

```
ALTER DATABASE Exemplo2
ADD FILEGROUP Grupo1_Exemplo2
GO
```

```
ALTER DATABASE Exemplo2
ADD FILE
(NAME=Teste,
FILENAME='C:\MSSQL7\DATA\TesteGrupo1_Exemplo2.ndf',
```

```

        SIZE=3MB)
    TO FILEGROUP Grupo1_Exemplo2
GO

```

Note que o parâmetro TO FILEGROUP simplesmente especifica o grupo em que o arquivo criado deve ser adicionado.

Agora, vamos usar o comando CREATE DATABASE para criar um arquivo de dados primário, um grupo de arquivos definido pelo usuário, e um arquivo de log. Então, emitiremos o comando ALTER DATABASE para mudar o grupo de arquivos padrão para o grupo de arquivos definido pelo usuário.

```

USE MASTER
GO
-- dois traços indicam um comentário
-- CRIAR O BANCO DE DADOS

CREATE DATABASE ExemploNovo
ON
PRIMARY (NAME=ExemploNovo_data,
FILENAME = 'c:\mssql7\data\ExemploNovo.mdf',
SIZE=10MB,
MAXSIZE=15MB,
FILEGROWTH=10%),
FILEGROUP ExemploNovo_FG1
(NAME=ExemploNovo_FG1_DAT1,
FILENAME='c:\mssql7\data\ExemploNovo_FG1_DAT1.ndf',
SIZE=3MB,
MAXSIZE=10MB,
FILEGROWTH=10%),
(NAME=ExemploNovo_FG1_DAT2,
FILENAME='c:\mssql7\data\ExemploNovo_FG1_DAT2.ndf',
SIZE=3MB,
MAXSIZE=10MB,
FILEGROWTH=10%)
LOG ON
(NAME=ExemploNovo_log,
FILENAME='c:\mssql7\data\ExemploNovo.ldf',
SIZE=5MB,
MAXSIZE=15MB,
FILEGROWTH=10%)
GO
-- Use ALTER para mudar o grupo de arquivos padrão
ALTER DATABASE ExemploNovo
MODIFY FILEGROUP ExemploNovo_FG1 DEFAULT
GO

```

Visualizando informações de arquivos e grupos de arquivos

Os seguintes procedimentos armazenados do sistema exibem informações sobre grupos de arquivos

- **sp_helpfile** [*file_name*]: exibe os nomes físicos e atributos dos arquivos associados com o banco de dados atual. Use este procedimento armazenado para determinar os nomes dos arquivos a serem anexados ou removidos do servidor.

- **sp_helpfilegroup** [*filegroup_name*]: exibe os nomes e atributos de grupos de arquivos associados com o banco de dados atual.

Criando Tabelas

Uma tabela [table] é um objeto do banco de dados, composto de zero ou mais *linhas* [rows], contendo os dados, organizados em uma ou mais *colunas* [columns]. Para criar a tabela, você pode usar o Enterprise Manager ou comandos SQL DDL (Data Definition Language - linguagem de definição de dados). Antes de criar suas tabelas, é importante levar em conta um bom projeto do banco de dados, que determina quais as informações a serem guardadas. Após criar as tabelas, você utiliza comandos SQL DML (Data Manipulation Language - linguagem de manipulação de dados) para inserir novas linhas numa tabela, alterar colunas das linhas existentes, excluir linhas e consultar dados.

Tipos de Dados

Cada coluna tem um *tipo de dados* [datatype], que determina que tipo de informação (caracteres, números, datas/horas) pode ser colocada na coluna e quais as características desses dados. O tipo é determinado quando a tabela é criada e não pode ser alterado posteriormente. Você pode usar *tipos de dados do sistema* [system datatypes], predefinidos, ou criar novos tipos de dados, chamados *tipos de dados do usuário* [user datatypes], baseados nos tipos preexistentes.

Os tipos de dados existentes são:

Para dados	Tipo	Tamanho
Caractere	<i>char</i> (n), <i>varchar</i> (n), <i>nvarchar</i> (n), <i>nchar</i> (n)	até <i>n</i> bytes
Numérico exato	<i>decimal</i> (p,e) ou <i>numeric</i> (p,e)	-depende-
Numérico aproximado	<i>float</i> , <i>real</i>	8, 4 bytes
Numérico inteiro	<i>int</i> , <i>smallint</i> , <i>tinyint</i>	4, 2, 1 byte
Monetário	<i>money</i> , <i>smallmoney</i>	8, 4 bytes
Data e hora	<i>datetime</i> , <i>smalldatetime</i>	8, 4 bytes
Binário	<i>binary</i> (n), <i>varbinary</i> (n)	<i>n</i> bytes
Texto e imagens	<i>text</i> , <i>image</i> , <i>ntext</i>	-variável-
Outros	<i>bit</i> , <i>timestamp</i>	1 bit, 8 bytes

Para dados contendo caracteres, *char*(n) armazena um número fixo de caracteres. Por exemplo, uma coluna do tipo *char*(30) tem sempre 30 caracteres. Se forem informados menos, o restante é completado com espaços. Já o tipo *varchar*(n) armazena uma quantidade variável de caracteres, até o máximo informado. Os tipos *nchar*(n) e *nvarchar*(n), armazenam dados Unicode, de comprimento fixo ou variável, e usam o conjunto de caracteres UNICODE UCS-2.

Os tipos "numéricos exatos", *decimal* e *numeric*, permitem armazenar dados exatos, sem perdas devidas a arredondamento. Ao usar esses tipos, você pode especificar uma *precisão*, que indica quantos dígitos podem ser usados no total e uma *escala*, que indica quantos dígitos podem ser usados à direita do ponto. Por exemplo, *decimal*(9,2) permite guardar 7 dígitos antes do ponto decimal e 2 após, num total de 9, assim o maior valor possível é 9999999,99.

Os tipos "numéricos inexatos", *float* e *real*, armazenam dados numéricos, mas nem sempre mantêm a precisão suficiente para armazenar corretamente números de vários dígitos.

O tipo *money* é usado para valores monetários, ocupando 8 bytes em disco e permitindo valores entre -922.337.203.685.477,5808 e +922.337.203.685.477,5807 (922 trilhões). O tipo *smallmoney* permite valores entre - 214.748,3648 e +214.748,3647 (214 mil) e ocupa 4 bytes em disco.

Dos tipos inteiros, *int* usa 32 bits (4 bytes), permitindo armazenar até +/-2.147.483.647, *smallint* usa 16 bits (2 bytes) permitindo +/-32767 e *tinyint* usa 8 bits (1 byte), permitindo números não-negativos de 0 a 255.

O tipo *datetime* armazena valores contendo a data e hora, com precisão de 1/300 de segundo, entre 1º de janeiro de 1753 e 31 de dezembro de 9999 (o século é sempre armazenado). O tipo *smalldatetime* ocupa menos espaço e armazena datas e horas de 1º de janeiro de 1900 até 6 de junho de 2079, com precisão de 1 minuto.

Tipos binários são usados para dados que o SQL Server não interpreta, por exemplo, o conteúdo de um arquivo binário. O tipo *text* é usado para colunas com dados "memo", ou seja, com texto de tamanho variável; o tipo *ntext* armazena dados Unicode de tamanho variável. O tipo *image* armazena imagens, também de tamanho variável.

Os tipos *text* e *ntext*, armazenam dados de tamanho variável, mas podem armazenar 1.073.741.823 caracteres, para o caso do *ntext*, e 2.146.483.647 caracteres para o caso do tipo *text*. Enquanto isso, os tipos *varchar* e *nvarchar* armazenam "somente" 8000 caracteres (*varchar*) ou 4000 caracteres (*nvarchar*)

O tipo *bit* armazena valor 1 ou 0. Uma coluna do tipo *timestamp* não pode ser alterada pelo usuário. Ela é definida automaticamente com a data e hora atual quando a linha é inserida ou atualizada.

Definindo novos tipos de dados

Você pode criar seus próprios tipos de dados, para facilitar a padronização, usando o procedimento de sistema *sp_addtype* ou o Enterprise Manager.

Usando o procedimento de sistema *sp_addtype*:

Sintaxe

```
sp_addtype nome_tipo, tipo_dado [, valor_null]
```

Onde:

nome_tipo é o nome do tipo de dado que deseja criar.

tipo_dado é o tipo de informação que ira conter este tipo criado. Exemplo: char, int,..etc.

valor_null identifica se este tipo pode ou não conter valores nulos.

Exemplos:

No Query Analyzer na lista "DB", selecione o banco de dados "Exemplo". Digite e execute os seguintes comandos:

```
sp_addtype cpf, 'char(11)'
go
sp_addtype nomepessoa, 'char(50)'
go
sp_addtype valorgrande, 'numeric(15,2)'
go
```



```
sp_addtype tipooperacao, 'SmallInt', NONNULL
```

A palavra reservada **go** indica final de comando.

Com isso, usar o tipo 'cpf', por exemplo, é o mesmo que usar char(11), mas é mais intuitivo e fácil de entender. Se você especificar NONULL no tipo significa que ele não aceita valores nulos.

Você pode excluir um tipo com *sp_droptype*:

Sintaxe

```
sp_droptype nome_tipo
```

Onde:

nome_tipo é o nome do tipo de dados que deseja excluir.

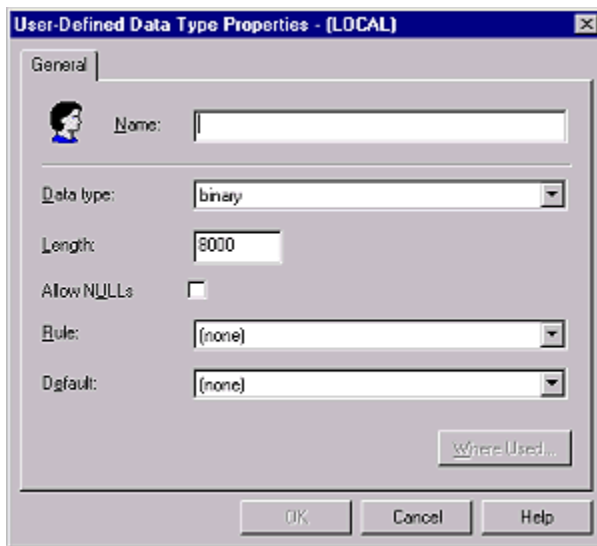
Exemplo:

Para executar este exemplo continue posicionado no banco de dados Exemplo.

```
sp_droptype nomepessoa
```

Para criar, alterar ou excluir tipos com o Enterprise Manager, você deve abrir *Nome-do-banco-de-dados* clique em User Defined Datatypes com o botão direito clique em

Refresh (para atualizar os dados) e com o botão direito clique em **New UserDefinedDataType....**



Será mostrada a seguinte tela:

'Name ' indica o nome do tipo de dados.

'Data type' é o tipo de informação que irá conter esse tipo criado.

'Length' indica o tamanho do tipo de dado.

'Allow Null' se esta opção estiver marcada indica que o tipo criado aceita valores nulos.

As opções 'Default' e 'Rule' permitem que você selecione uma regra ou um default, se houver algum, e ligue-o ao tipo de dados definido por você.

Para apagar algum tipo de dados, selecione-o do lado direito (quando você estiver com User Defined DataTypes selecionado do lado esquerdo do Enterprise Manager). Clique

com o botão direito no tipo de dados que você quer excluir, e então selecione a opção Delete.

Apague todos os tipos de dados e crie o tipo de dados Sexo. Em "Name" coloque dmSexo, "Data Type" selecione char, "Length" coloque 1. Mais tarde veremos como restringir o valor de um tipo.

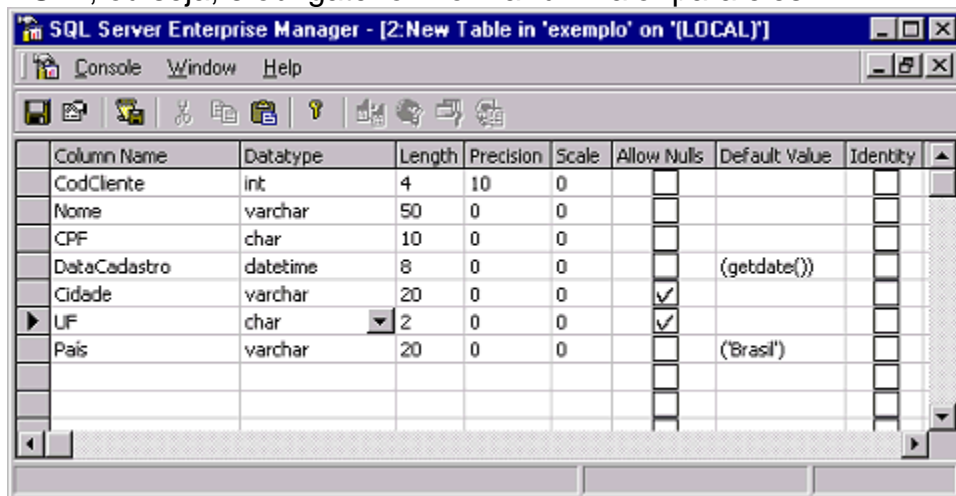
Criando uma tabela com o Enterprise Manager

Um banco de dados pode ter no máximo 2 bilhões de tabelas e cada tabela pode ter no máximo 1024 colunas.

Para criar uma tabela com o Enterprise Manager, abra o banco de dados Exemplo. Dentro dele, selecione o item Tables. Clique em "Tables" com o botão direito e em **New Table**. Entre com o nome da tabela a ser criada. No caso Cliente; clique em Ok. Entre com os campos da tabela, conforme mostrado abaixo:

Veja que no título da janela, aparece o nome do banco de dados em que a tabela está sendo criada (no caso 'exemplo').

As colunas para as quais a opção "Nulls" está marcada permitem o valor NULL, ou seja, podem ser deixados sem preencher ao inserir dados. Já os outros são NOT NULL, ou seja, é obrigatório informar um valor para eles.



Note que definimos algumas colunas com o tipo *char*, como CPF e UF, porque elas geralmente têm tamanho fixo. Já outras como Nome, Cidade e País, geralmente têm tamanho variável, por isso, para economizar espaço no banco de dados, usamos *varchar*.

A coluna Default especifica um valor default que é inserido caso nada tenha sido informado. No caso da data de cadastro, usamos a função `getdate()`, que retorna a data do dia. No caso do país, o default é a string "Brasil" caso nada seja informado.

Agora clique no botão "Save" para salvar a tabela.

Nota: Você pode ter um identificador exclusivo em todo o banco de dados, para uma coluna. Para isso, selecione o tipo de dados da coluna como **uniqueidentifier**, e

marque a caixa de verificação IsRowGuid. Isso fará com que seja atribuído um valor default [Default Value] igual a (newid()). Veja mais sobre identificadores globalmente exclusivos.

Criando tabelas com comandos SQL

Uma tabela também pode ser criada com o comando CREATE TABLE do SQL. Por exemplo, a mesma tabela do exemplo anterior poderia ser criada com o comando abaixo, iremos mudar somente o nome da tabela e esta tabela será criada no banco de dados Exemplo:

```
CREATE TABLE Cliente1
(
  CodCliente int NOT NULL,
  Nome varchar(50),
  CPF varchar(11) NULL,
  DataCadastro datetime NOT NULL DEFAULT (getdate()),
  Cidade varchar(20) NULL,
  UF char(2) NULL,
  País varchar(20) DEFAULT ('Brasil')
)
```

Note que a lista de colunas é delimitada com parênteses. Para cada coluna, deve-se informar NULL ou NOT NULL, indicando se esta permite valores nulos ou não. Caso essa opção não seja informada, como no caso de 'Nome' e 'País' acima, o SQL Server assume que a coluna é NOT NULL (geralmente).

Para saber se o SQL usa NULL ou NOT NULL por default execute o procedimento **sp_dboption**. Se aparecer na primeira linha "ANSI null default" indica que o valor default é NULL, se não aparecer esta string a opção default é NOT NULL.

Sintaxe

```
sp_dboption nome_bancodedados 'ansi null default', opcao
```

Onde:

nome_bancodedados é o nome do banco de dados em que se deseja verificar a opção default.

opcao se for true o valor default será NULL, se for false o valor default será NOT NULL.

Observação: O padrão ANSI utiliza Null como default, ao desativar esta opção no SQL, pode ser que esta opção na sua tabela não esteja de acordo com o que você esperava.

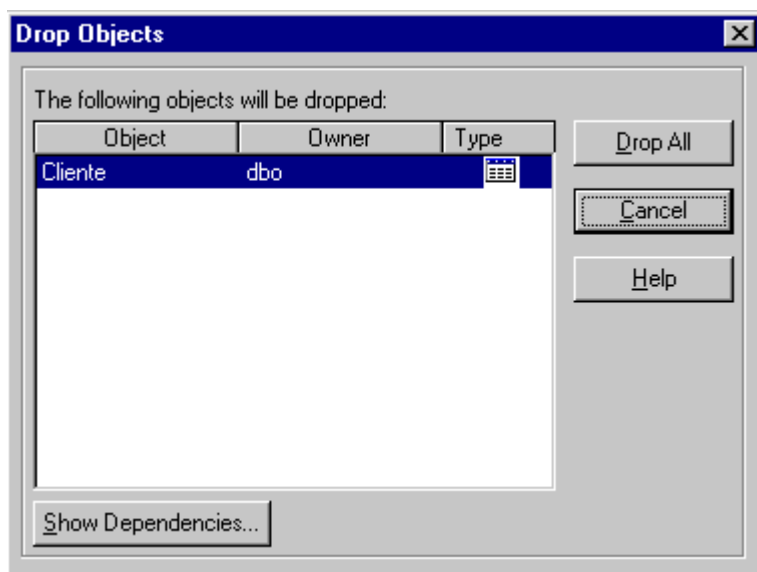
Veja outros parâmetros de sp_dboption em "Definindo opções do banco de dados"

Execute os seguintes comandos para criar duas tabelas no banco de dados Exemplo:

```
CREATE TABLE Departamento
(
  CodDepartamento int,
  Nome varchar(50),
  CodDeptSuperior int Null
)
go
```

```
CREATE TABLE Funcionario
(
  CodFuncionario int,
  Nome varchar(50),
  CodDepartamento int,
```

```
Ramal int Null,  
Salario money,  
DataAdmissao datetime,  
DataCadastro datetime NOT NULL DEFAULT (getdate()),  
Sexo char(1)  
)
```



Excluindo uma tabela

Para excluir uma tabela (chamado de *drop* no SQL Server) com o Enterprise Manager, clique na tabela com o botão direito, clique em **Delete** e pressione o botão Drop All .

Como iremos usar a tabela Cliente nos exemplos posteriores , crie-a novamente com a mesma estrutura definida anteriormente.

Através do SQL, pode-se usar o comando DROP TABLE.

Sintaxe:

```
DROP TABLE [[banco_dados.]Owner.]nome_tabela  
[, [[banco_dados.]owner.]nome_tabela...]
```

Onde:

banco_dados é o nome do banco de dados a que a tabela pertence. Essa opção é opcional, ela será usada somente quando se estiver posicionado num determinado banco de dados e se deseja excluir a tabela de outro banco de dados.

nome_tabela é o nome da tabela que se deseja remover.

Este procedimento pode ser usado para excluir várias tabelas ao mesmo tempo.

Exemplos:

```
Drop Table Clientel
```

ou

```
Drop Table Exemplo.dbo.clientel
```

Alterando a estrutura das tabelas

Depois que uma tabela for criada, pode-se mudar várias das opções que foram definidas quando a tabela foi originalmente criada, incluindo:

- Colunas podem ser acrescentadas, modificadas ou excluídas. Por exemplo, o nome da coluna, comprimento, tipo de dados, precisão, escala, e o fato de aceitar ou não valores nulos, podem todos ser mudados, embora existam algumas restrições.
- Restrições PRIMARY KEY e FOREIGN KEY podem ser acrescentadas ou excluídas.
- Restrições UNIQUE e CHECK e definições DEFAULT podem ser acrescentadas ou excluídas.
- Uma coluna identificadora pode ser acrescentada ou removida usando a propriedade IDENTITY ou ROWGUIDCOL. A propriedade ROWGUIDCOL também pode ser adicionada ou removida de uma coluna existente, embora apenas uma coluna em uma tabela possa ter a propriedade ROWGUIDCOL de cada vez.

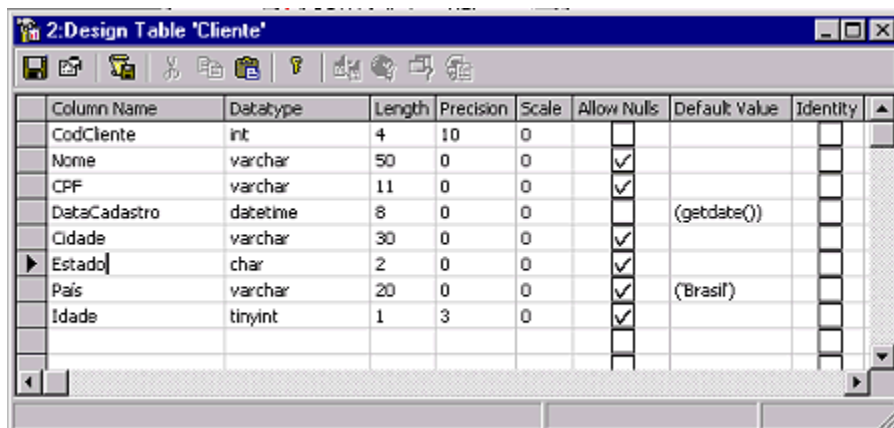
O nome ou o dono de uma tabela também podem ser modificados. Quando você faz isso, também deve-se mudar o nome da tabela em quaisquer gatilhos, procedimentos armazenados, scripts SQL, ou outro código de programação que utilize o nome ou proprietário antigo da tabela.

Nota: É importante considerar que a mudança de tipo de dados em uma coluna pode causar truncamento dos dados, ou mesmo ser impossível de ser feita (por exemplo, se você quiser converter um tipo char para um tipo inteiro e já houver valores não-numéricos armazenados nessa coluna).

Alterando a tabela com o Enterprise Manager

Essas modificações podem ser feitas no Enterprise Manager. Por exemplo, clique na tabela "Cliente" (dentro de "Exemplo\Tables", as tabelas do banco de dados Exemplo aparecerão do lado direito do Enterprise Manager) com o botão direito e clique em **Design Table**. No final da lista de colunas, acrescente uma nova coluna, com o nome "Idade", do tipo "tinyint" (idades não serão maiores que 255). Note que quando você adiciona uma nova coluna, a opção "Allow Nulls" deve ficar marcada.

Altere o comprimento de Cidade para 30. Também clique na coluna "UF" e altere o



nome para "Estado". Após fazer isso, clique no botão "Save" para atualizar a tabela.

Alterando a tabela com comandos SQL

Também é possível alterar uma tabela com comandos SQL. Para isso, use o comando ALTER TABLE. Abaixo será mostrada uma sintaxe simples deste procedimento:

Sintaxe:

```
ALTER TABLE [banco_dados.[owner.]]nome_tabela
```

```
{
[ALTER COLUMN nome_coluna
    {novo_tipo_de_dados [(precisão[, escala])]}
| ADD {nome_coluna dados_coluna
    | [WITH CHECK | WITH NOCHECK]}
}
```

Onde:

banco_dados é o nome do banco de dados a que a tabela pertence. Essa opção é opcional, ela será usada somente quando se estiver posicionado num determinado banco de dados se e deseja excluir a tabela de outro banco de dados.

nome_tabela é o nome da tabela que deseja alterar.

nome_coluna é a coluna que se quer alterar.

novo_tipo_de_dados é o tipo de dados que a coluna aceitará a partir de agora.

WITH CHECK | WITH NOCHECK Especificam se os dados na tabela devem ou não ser validados contra uma nova ou reabilitada restrição FOREIGN KEY ou CHECK. Se não especificada, assume-se WITH CHECK para novas restrições e WITH NOCHECK para restrições reabilitadas. Veja mais sobre restrições.

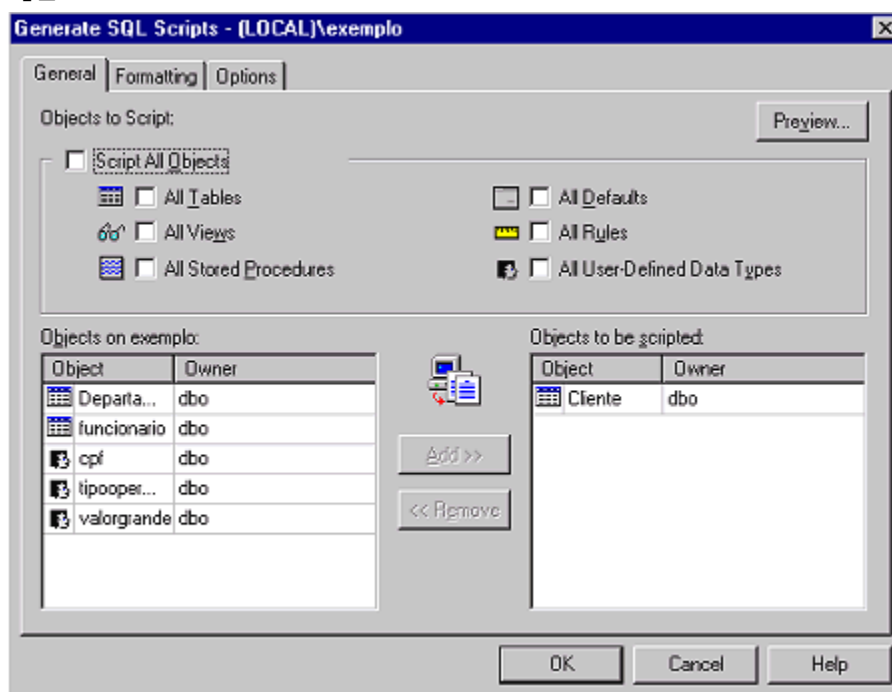
Exemplo:

```
ALTER TABLE Cliente
ADD ender varchar(50) NULL
GO
ALTER TABLE Cliente
```

```
ALTER COLUMN CIDADE VARCHAR (25)
```

Para renomear uma coluna, usa-se o procedimento *sp_rename*:

```
sp_rename 'Cliente.ender', Endereco
```



Nota: Perceba que ao executar o comando acima, você recebe um aviso que diz: "Cuidado: Mudar qualquer parte do nome de um objeto pode invalidar scripts e procedimentos armazenados."

Criando um "Script" das tabelas

Algumas tarefas são mais fáceis de fazer com os comandos CREATE TABLE. Por exemplo, para recriar um banco de dados em outro servidor, você pode salvar um arquivo (um *script*) contendo todos os comandos SQL CREATE TABLE usados para criar suas tabelas. Um script em geral é um arquivo contendo comandos SQL.

Se você não usou um comando SQL, pode fazer o próprio SQL Server gerar um script para você a partir da tabela existente. Para isso, no Enterprise Manager, clique no nome da tabela com o botão direito, e em "All tasks", e depois em "Generate SQL Scripts". Por enquanto, deixe as opções default. Clique em "Preview" para ver como ficam os comandos. Depois clique em "Save As..." para salvar o arquivo Script. Salve o arquivo com o nome Cliente (a terminação .SQL já é colocada por padrão).

Feche as janelas.

Para executar esse script em outro servidor ou outro banco de dados, basta abrir o SQL Query Analyzer (Iniciar | Programas | Microsoft SQL Server 7.0 | Query Analyzer), abrir o arquivo de script e executá-lo. Para abrir o arquivo Script clique em e procure o nome do arquivo. Note que você pode também modificar os comandos do script para criar tabelas com colunas ligeiramente diferentes.

Também é possível criar um script com todas as tabelas do banco de dados, ou com todos os objetos. Para isso, dê uma olhada em "Documentando a criação de bancos de dados" (veremos outros tipos de objetos mais tarde).

Definindo opções de bancos de dados

Uma porção de opções de bancos de dados podem ser definidas para cada banco de dados. Apenas o Administrador de Sistema (SA) ou o proprietário do banco de dados pode mudar estas opções. A mudança destas opções só modificará o banco de dados atual; não afetará outros bancos de dados.

As opções de bancos de dados podem ser modificadas com o procedimento armazenado de sistema **sp_dboption**, ou através do Enterprise Manager. O procedimento armazenado **sp_dboption** só afeta o banco de dados atual, mas para modificar opções a nível de servidor, use o procedimento armazenado de sistema **sp_configure**.

Depois de fazer alguma mudança, é emitido automaticamente um checkpoint, de modo que as mudanças são imediatas.

Opções disponíveis

A seguir, temos uma lista das opções mais comuns de banco de dados. Para maiores detalhes em cada uma das opções, veja no Books Online.

As opções marcadas com um asterisco (*) indicam que essa opção pode ser configurada pelo Enterprise Manager; caso contrário, é uma opção só alterável através de procedimentos armazenados.

***ANSI null default**

Controla se o valor padrão para todos os tipos de dados é NULL. A Microsoft põe o padrão em NOT NULL. Se esta opção estiver em TRUE, o padrão será NULL para o banco de dados. Quando se entrar com o comando CREATE TABLE, a não ser que o

criador indique explicitamente NOT NULL, a regra se aplicará também à criação da tabela.

ANSI Nulls

Quando em TRUE, as comparações de NULL com qualquer valor vão retornar um NULL. Quando em FALSE, apenas comparação de valores não-Unicode retornarão TRUE se e somente se ambos valores forem nulos. O padrão para essa opção é FALSE.

ANSI Warnings

Quando em TRUE, avisos de erro são exibidos, quando ocorrerem condições tais como divisão por zero ou valores nulos aparecerem em funções de agregação. Por padrão, é FALSE.

***autoclose**

Quando em TRUE, o banco de dados é fechado automaticamente quando o último usuário encerra a conexão. Isto é muito útil para ambientes pequenos, mas deve ser evitado nos casos em que conexões são constantemente feitas e encerradas. A quantidade de carga adicional gerada pela abertura e fechamento de um banco de dados pode ter efeitos negativos em um ambiente de produção.

autoshrink

Quando em TRUE, o SQL Server periodicamente reduzirá os arquivos do banco de dados se necessário.

***dbo use only**

Quando em TRUE, apenas o dbo (proprietário do banco de dados) tem acesso ao banco de dados. Use esta opção quando estiver executando reparos nem bancos de dados.

published

Utilizado para relicação, quando *published* estiver em TRUE, indica que a publicação está habilitada. Colocar essa opção em FALSE desabilita a publicação.

***read only**

Se TRUE indica que o banco de dados é somente para leitura. FALSE permite acesso para leitura/escrita.

***recursive triggers**

Quando TRUE, é permitido o disparo de gatilhos recursivos [recursive triggers]. Quando FALSE (o padrão), gatilhos não podem disparar recursivamente. Um gatilho recursivo é aquele que dispara na tabela que o originou, causando uma atualização em outra tabela, a qual causa uma atualização na tabela que originou o gatilho.

***selec into / bulk copy**

Permite que o banco de dados aceite ações não registradas em log, tais como SELECT INTO e o utilitário BCP fazem.

***single user**

Permite que apenas um usuário acesse o banco de dados.

subscribed

Quando em TRUE, o banco de dados pode ser assinado para publicação.

***torn page detection**

Se TRUE, o SQL Server detectará leituras incompletas em disco, e fará com que sejam marcadas. Quedas de energia ou outros defeitos podem causar essas leituras incompletas.

Truncate log on Checkpoint (*trunc. Log on chkpt.)

Quando estiver em TRUE, o SQL Server trunca o log de transações toda vez que encontrar um checkpoint. Esta opção é usada frequentemente para desenvolvimento, fazendo com que o log de transações não fique cheio com tanta frequência. Você não deve utilizar esta opção em um sistema "real".

Definindo opções do banco de dados com sp_dboption

Para mudar as opções de um banco de dados com o procedimento armazenado sp_dboption, faça o seguinte:

Sintaxe:

```
sp_dboption ['banco_de_dados'] [, 'opção'] [, 'valor']
```

Por exemplo:

```
sp_dboption 'pubs', 'read only', 'true'
```

Para ver o estado atual das opções do banco de dados **pubs**, entre com o seguinte comando:

```
sp_dboption 'pubs'
```

Todas as opções que estiverem ativadas são listadas.

Definindo opções do banco de dados pelo Enterprise Manager

Quando se utiliza o Enterprise Manager para configurar as opções do banco de dados, você só tem acesso a um subconjunto (cerca de metade) das opções realmente disponíveis.

Para mudar opções do banco de dados com o Enterprise Manager, faça assim:

1. Expanda o grupo do servidor.
2. Expanda o servidor.
3. Expanda os bancos de dados.
4. Clique com o botão direito no banco de dados que você quer mudar, e então clique em Propriedades [Properties].
5. Selecione as opções a mudar.
6. Clique em OK quando tiver acabado.

Verificando propriedades do banco de dados

A seguir você vê alguns procedimentos armazenados de sistema, frequentemente utilizados, que exibem informações sobre bancos de dados e opções de bancos de dados.

- **sp_dboption**: como visto acima, mostra todas as opções disponíveis para o banco de dados em que se estiver posicionado.
- **sp_helpdb**: informações sobre todos bancos de dados em um servidor. Fornece nome do banco de dados, tamanho, proprietário, ID, data de criação, e opções.
- **sp_helpdb nome_banco_de_dados**: informações sobre um banco de dados específico apenas. Fornece nome do banco de dados, tamanho, proprietário, ID, data de criação, e opções. Além disso, lista os arquivos para dados e log de transações.
- **sp_spaceused [nome_objeto]**: resumo do espaço de armazenamento que um banco de dados, log de transações, ou objeto de banco de dados utiliza.

Considerações para melhor gerenciamento

Para que você possa trabalhar com mais tranquilidade e eficiência com bancos de dados, considere os seguintes fatos.

- Para obter melhor desempenho e segurança, armazene o banco de dados e o log de transações em discos físicos separados.
- Desabilite o cache de escrita nos controladores de disco, a menos que o mecanismo de cache de escrita seja especificamente projetado para servidores de bancos de dados.
- Faça backup do banco de dados master imediatamente depois de criar ou modificar bancos de dados. Em geral, é uma boa idéia fazer backup dos bancos de dados regularmente.
- Garanta que você tenha espaço suficiente para o log de transações. Se você ficar sem espaço, você não será capaz de modificar ou acessar seu banco de dados. Para evitar ficar sem espaço, faça o seguinte:
 - Aloque espaço suficiente para acomodar o crescimento.
 - Monitore frequentemente o espaço total sendo usado.
 - Use a opção de crescimento automático para aumentar o espaço em disco automaticamente.
 - Configure um alerta para te avisar quando o espaço disponível no log de transações esteja abaixo de 25 por cento do espaço total do log de transações.

Exclusão de bancos de dados

Você não pode excluir bancos de dados que estejam:

- Atualmente abertos para leitura ou escrita por outro usuário.
- Sendo restaurados.
- Publicando qualquer de suas tabelas (parte da replicação SQL).

Você também não pode excluir os seguintes bancos de dados:

- Master
- Model

- Tempdb

Embora lhe seja permitido excluir o banco de dados de sistema msdb, você não deve excluí-lo se usa ou pretende usar:

- Replicação
- SQL Server Agent
- Assistente de criação de páginas Web
- Histórico de backups
- Serviços de transformação de dados

Quando excluir um banco de dados, considere os seguintes fatos:

- Com o método SQL DROP DATABASE, você pode excluir vários bancos de dados de uma vez.
- O Enterprise Manager só lhe permite excluir um banco de dados de cada vez.
- Depois que você excluir um banco de dados, qualquer ID de login que usava o banco de dados excluído como seu banco de dados padrão, usará agora o banco de dados **master**.
- Você deve SEMPRE fazer backup do banco de dados **master**, sempre que qualquer novo banco de dados for adicionado ou excluído.

Documentação dos passos de criação de bancos de dados no SQL Server

Documentar os passos de criação de bancos de dados SQL Server pode ser útil por diversas razões, mas é claro que a principal e mais motivadora é o fato de ter um backup do trabalho que você fez. Isso não necessariamente vai lhe prevenir de perdas de dados, mas vai salvar seu modelo, e um modelo de banco de dados é uma coisa terrível a se perder.

O SQL Server tem um gerador de script que torna fácil para você documentar, e se necessário reconstruir, seu banco de dados. O gerador de scripts pode construir o banco de dados e os objetos criados no banco de dados. Você tem a opção de selecionar desde todos até um único objeto. Você pode pegar um script de um banco de dados e rodá-lo em outro para criar cópias exatas de procedimentos armazenados, regras, gatilhos, etc. Você pode gerar scripts para os seguintes objetos:

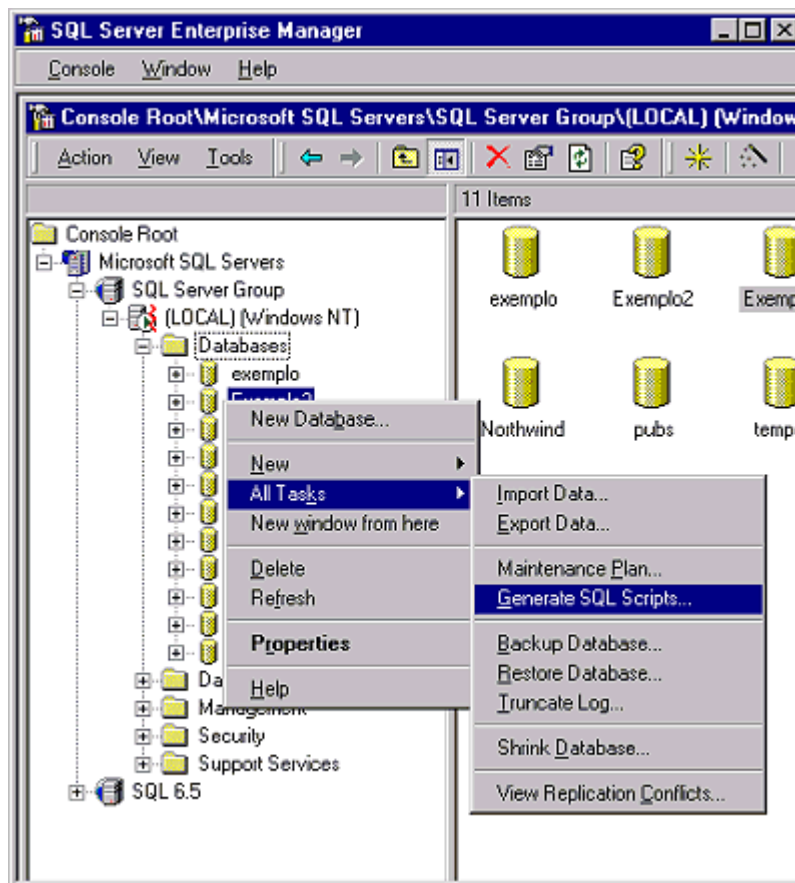
- Tabelas [Tables]
- Procedimentos armazenados [Stored procedures]
- Gatilhos [Triggers]
- Índices [Indexes]
- Visões [Views]
- Usuários e Grupos [Users and Groups]
- Tipos de dados definidos pelo usuário [User-defined data types]
- Logins
- Regras [Rules]

- Default
- Tabelas-chave / DRI

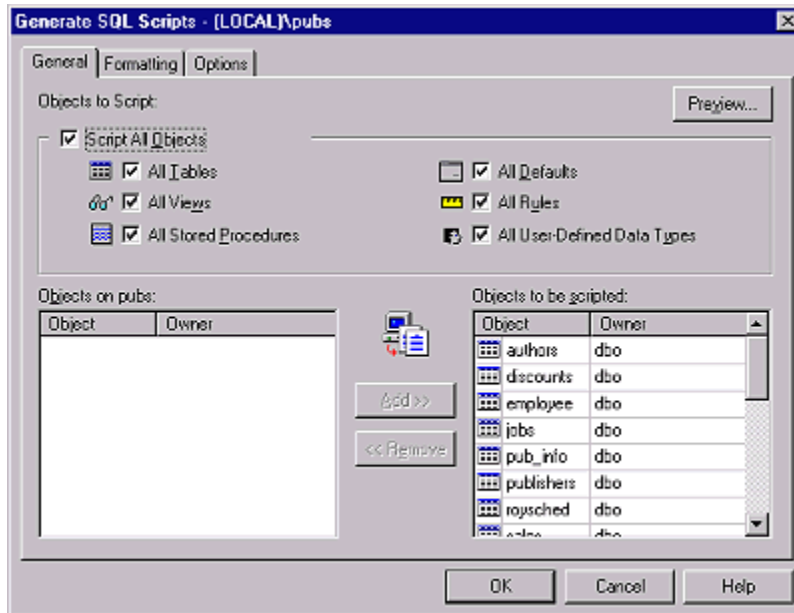
O esquema pode ser salvo em um arquivo único ou você pode querer dividi-lo baseado em objetos. Independentemente do seu método, você não tem mais uma boa desculpa para um banco de dados não documentado.

Gerando um script a partir do Enterprise Manager

1. Expanda o grupo de servidores.
2. Expanda o servidor.
3. Expanda os bancos de dados.
4. Clique com o botão direito no banco de dados escolhido, e selecione All Tasks.
5. Selecione Gerar Scripts SQL [Generate SQL Scripts], como indicado abaixo.



6. Selecione os objetos que você deseja criar da janela que aparece a seguir, mostrada abaixo:



Você pode prever o arquivo primeiro ou simplesmente clicar em OK para salvá-lo em um arquivo.

6 - Consultando Dados

Visão Geral do Transact-SQL

A Sintaxe do SELECT

Manipulando Expressões

Condições de Pesquisa

Outros Recursos

Objetivos:

- Entender a divisão de comandos da linguagem Transact-SQL;
- Aprender a usar o comando SELECT e suas várias opções para fazer consultas aos dados.

A Sintaxe do SELECT

O comando SELECT recupera dados de uma ou mais tabelas. A sua sintaxe mais simples pode ser resumida da forma:

Sintaxe:

```
SELECT lista_de_colunas  
FROM lista_de_tabelas  
WHERE condições
```

Onde

A *lista_de_colunas* especifica quais colunas serão retornadas como resultado, separadas por vírgulas ou um asterisco (*) que indica todas as colunas da tabela.

A cláusula FROM, com uma *lista_de_tabelas*, especifica quais tabelas serão consultadas.

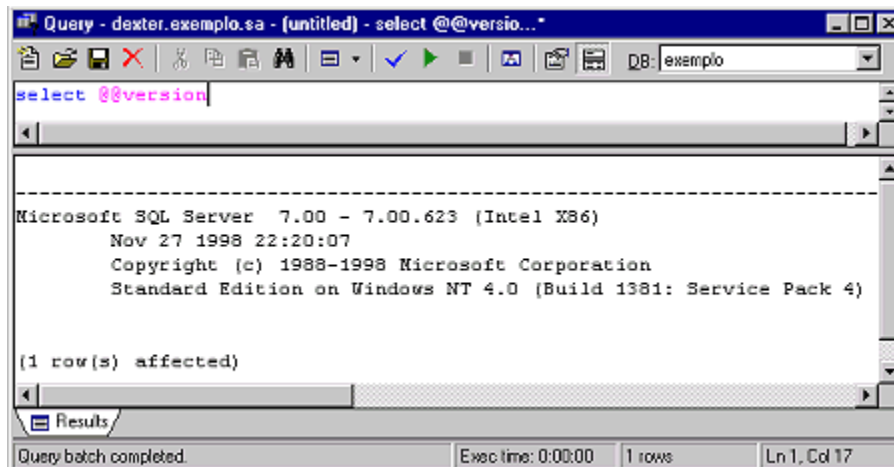
A cláusula WHERE especifica *condições* que devem ser satisfeitas pelas linhas das tabelas.

O comando Select pode ser utilizado para mostrar o conteúdo de variáveis, valores literais, etc...

Como exemplo execute o seguinte comando no Query Analyzer:

```
Select @@VERSION
```

O resultado será:



@@Version é uma variável global do SQL Server que contém a versão do SQL Server utilizado.

Para mostrar um valor literal digite o comando:

```
Select 'Teste'
```

O resultado será a palavra Teste.

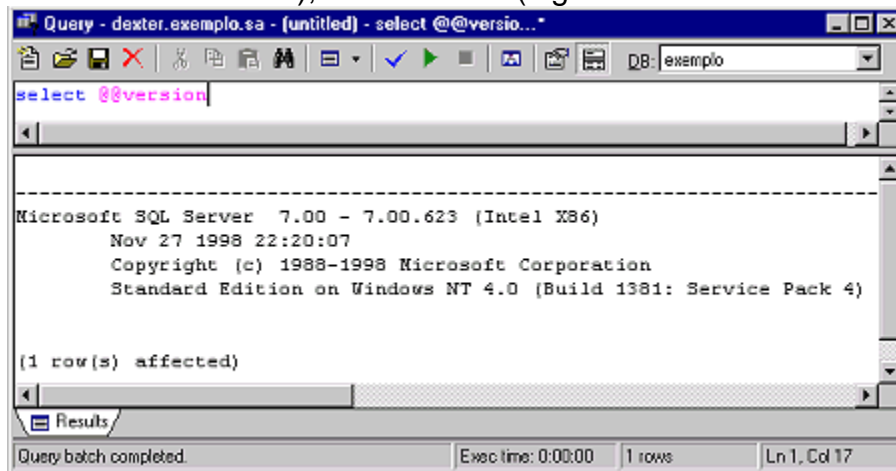
Exemplo: Consultando Todas as Colunas

Para nossos exemplos, vamos usar o banco de dados *pubs*, um banco de exemplo que é instalado pelo SQL Server. Esse banco de dados armazena informações sobre livros, na tabela *titles* e sobre autores de livros, na tabela *authors* (entre outras coisas).

Execute o Query Analyzer e se conecte ao servidor como "sa". Na lista "DB", selecione "pubs". Digite e execute o seguinte comando:

```
select * from authors
```

O resultado irá mostrar todas as colunas e todas as linhas da tabela 'authors' (ou seja, todo o seu conteúdo), como abaixo (algumas linhas e colunas foram omitidas):



Note a mensagem "23 row(s) affected" [23 linhas afetadas]. Isso indica quantas linhas foram retornadas pelo SELECT.

Você poderia escolher ver os resultados em uma grade, ao invés de em modo texto.

Para isso basta selecionar, clicando na seta à direita no ícone, a opção "Results in Grid" (Ctrl+D).



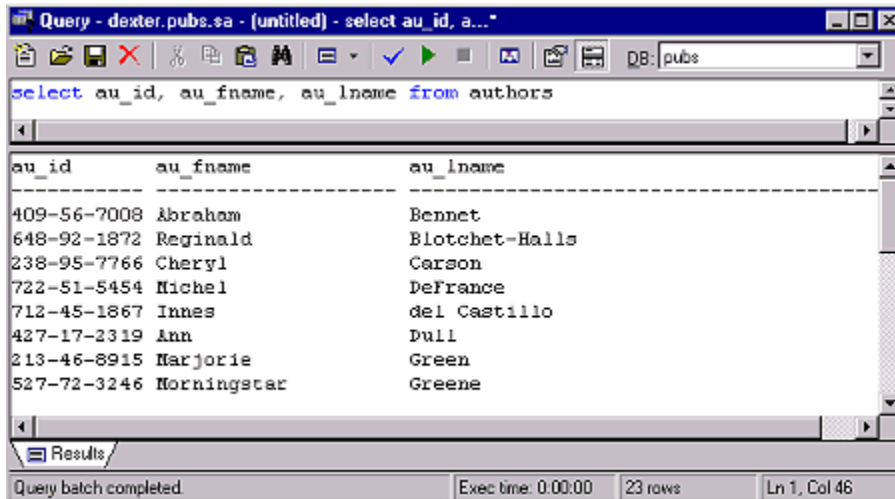
Caso você não queira selecionar na lista 'DB' o banco de dados a que a tabela que deseja procurar pertence, será necessário indicar no comando a qual banco de dados a tabela pertence, e o comando seria o seguinte:


```
select * from pubs..authors
```

Nessa tabela de exemplo, a coluna 'au_fname' é o primeiro nome do autor, 'au_lname' é o sobrenome [last name] e 'au_id' é o número de identificação. Agora suponhamos que você quer consultar apenas essas três primeiras colunas, omitindo a informação de telefone (phone) e endereço (address).

O '*' no comando acima especifica que todas as colunas da tabela são retornadas, mas você pode listar só as que são desejadas. Clique na página Query, e altere o comando anterior para o seguinte:

```
select au_id, au_fname, au_lname from authors
```



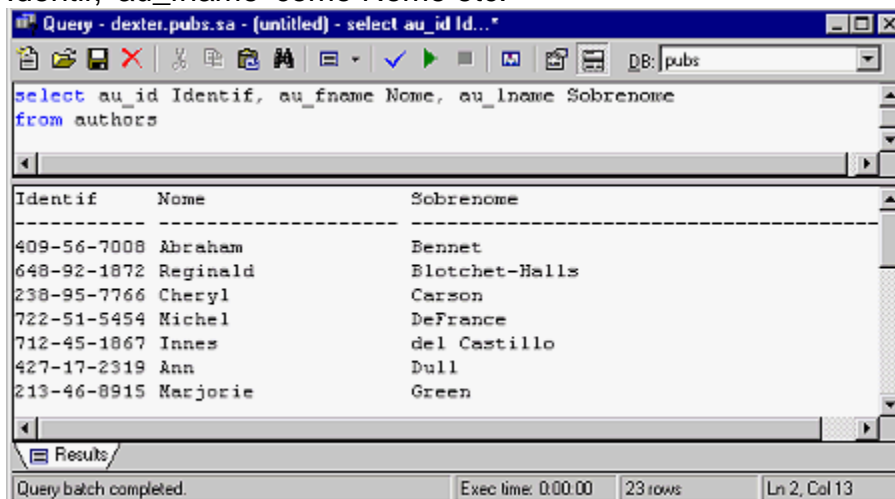
Execute o comando. Agora apenas as colunas 'au_id', 'au_fname' e 'au_lname' são retornadas, nessa ordem:

Note que a ordem das colunas não precisa ser a mesma ordem presente na definição da tabela. De fato, na maioria das aplicações bem construídas, a ordem das colunas na tabela não tem a menor importância.

Você também pode mudar o cabeçalho das colunas retornadas, criando um *alias* de coluna. Execute o seguinte comando:

```
select au_id Identif, au_fname Nome, au_lname Sobrenome
from authors
```

O resultado será o mesmo do comando anterior, mas a coluna 'au_id' aparece como Identif, 'au_fname' como Nome etc.



A palavra reservada **as** pode ser utilizada para indicar um alias, mas é opcional. Por exemplo:

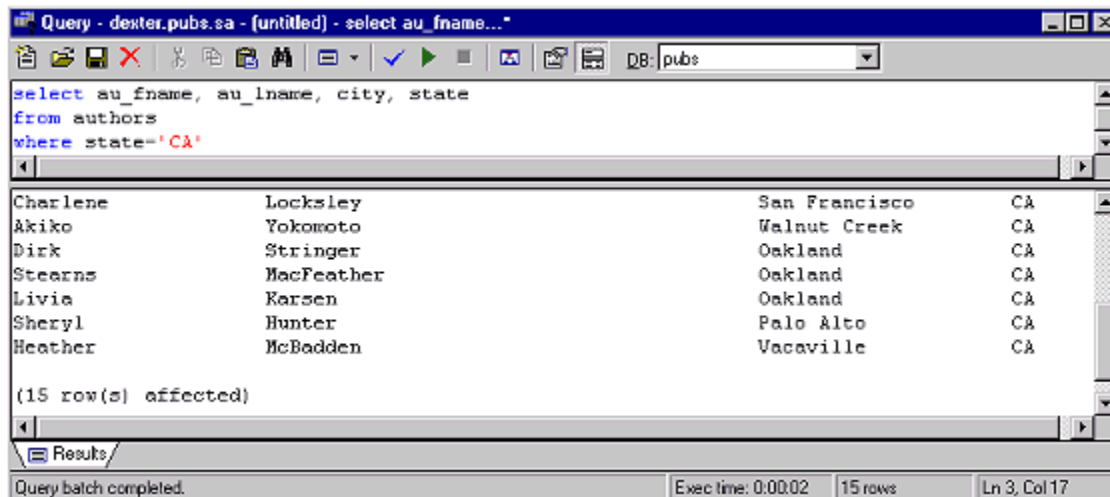
```
select au_id as Identif, au_fname as Nome, au_lname as Sobrenome
from authors
```

Usando Condições

Os comandos que já usamos não têm a cláusula WHERE. Nesse caso, todas as linhas da tabela são retornadas. Se o WHERE estiver presente, ele especifica uma condição que seleciona as linhas, e apenas as que satisfazem essa condição serão mostradas. Por exemplo, se quisermos os autores que moram na Califórnia, podemos consultar as linhas cuja coluna 'state' (estado) tem o valor 'CA':

```
select au_fname, au_lname, city, state
from authors
where state='CA'
```

O resultado será:



The screenshot shows a query window titled "Query - dexter.pubs.sa - (untitled) - select au_fname...". The query text is:


```
select au_fname, au_lname, city, state
from authors
where state='CA'
```

 The results are displayed in a table with 4 columns: au_fname, au_lname, city, and state. There are 15 rows of data. Below the table, it says "(15 row(s) affected)". At the bottom, a status bar indicates "Query batch completed.", "Exec time: 0:00:02", "15 rows", and "Ln 3, Col 17".

au_fname	au_lname	city	state
Charlene	Locksley	San Francisco	CA
Akiko	Yokomoto	Walnut Creek	CA
Dirk	Stringer	Oakland	CA
Stearns	MacFeather	Oakland	CA
Livia	Karsen	Oakland	CA
Sheryl	Hunter	Palo Alto	CA
Heather	McBadden	Vacaville	CA

Note que o resultado mostra apenas 15 linhas (15 rows affected) e não 23, que é o total da tabela. As linhas que aparecem são apenas as que satisfazem a consulta. Existem vários tipos de condições de pesquisa, como veremos.

Manipulando expressões

Um comando SELECT pode retornar nas colunas de resultado uma coluna da tabela, ou um valor calculado. Por exemplo, a tabela *titles* contém os títulos de livro (title) e os preços de cada um (price). Se quisermos ver como fica o preço de cada um após um aumento de 10%, pode ser feito o seguinte:

```
select price Preço, (price * 1.1) "Preço com 10% de
aumento", title from titles
```

Note que "Preço com 10% de aumento" é o nome do cabeçalho da expressão (`price * 1.1`), como o nome colocado possui espaços, foi necessário colocá-lo entre aspas.

Ou seja, a segunda coluna, cujo nome é "Preço com 10% de aumento" mostra o resultado de `price * 1.1` para cada linha. Você pode também usar vários operadores em expressões com colunas numéricas: adição (+), subtração (-), multiplicação (*), divisão (/) e módulo (%). O módulo só pode ser usado com tipos inteiros e calcula o resto da divisão de dois números inteiros (Ex.: $13 \% 4 = 1$).

Funções matemáticas

Além de operadores, você pode usar funções matemáticas do SQL Server, por exemplo:

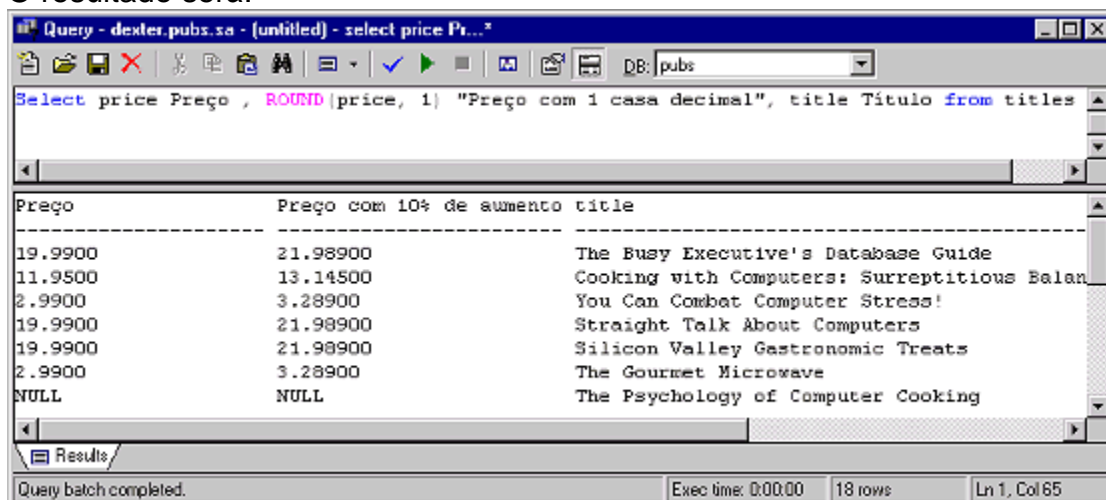
`ABS(valor)` retorna o valor absoluto (sem sinal) de um item.
`POWER(valor,p)` retorna o *valor* elevado à potência *p*.
`ROUND(valor,n)` arredonda o *valor* para *n* casas decimais.
`SQRT(valor)` retorna a raiz quadrada do valor especificado.
`PI` valor constante 3.141592563589793

Para outras funções, consulte o help do Transact-SQL em 'Math functions'.

Por exemplo, para arredondar o valor do preço de cada livro para duas casas decimais, pode ser feito o seguinte:

```
Select price Preço , ROUND(price, 1) "Preço com 1 casa decimal", title Título from titles
```

O resultado será:



Preço	Preço com 10% de aumento	title
19.9900	21.98900	The Busy Executive's Database Guide
11.9500	13.14500	Cooking with Computers: Surreptitious Balan
2.9900	3.28900	You Can Combat Computer Stress!
19.9900	21.98900	Straight Talk About Computers
19.9900	21.98900	Silicon Valley Gastronomic Treats
2.9900	3.28900	The Gourmet Microwave
NULL	NULL	The Psychology of Computer Cooking

Query batch completed. Exec time: 0:00:00 18 rows Ln 1, Col 65

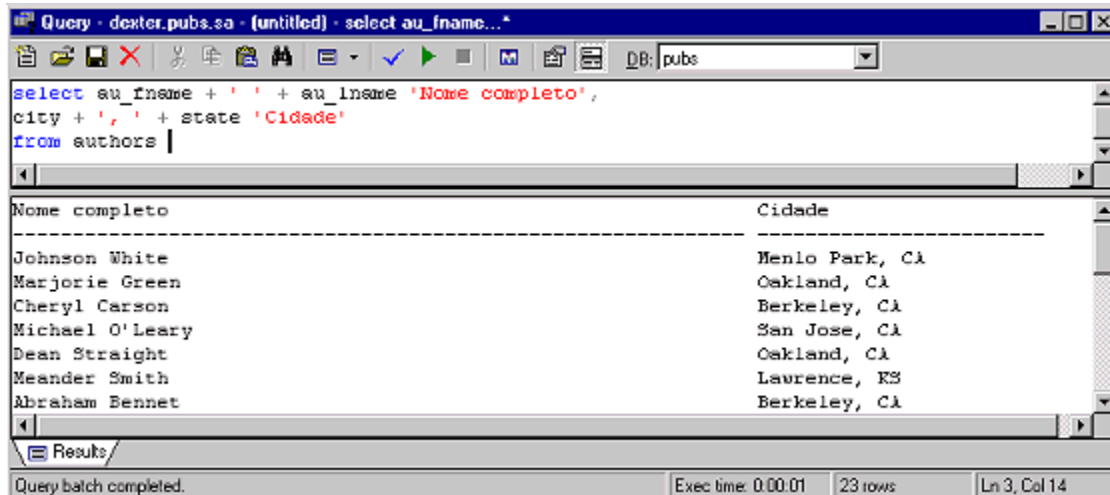
Funções de caracteres

Você pode usar funções para manipular dados do tipo caracter (*char* ou *varchar*), por exemplo, para pegar uma sub-string de uma seqüência de caracteres. E você pode usar o operador + para concatenar dois valores de tipo caracter.

Por exemplo, digite o seguinte comando:

```
select au_fname + ' ' + au_lname 'Nome completo',  
city + ', ' + state 'Cidade'
```

from authors



O resultado será:

O que fizemos foi concatenar o primeiro nome e segundo nome do autor, inserindo um espaço no meio (`au_fname + ' ' + au_lname`), para gerar a coluna Nome completo e depois juntar o nome da cidade com o nome do estado, inserindo uma vírgula (`city + ', ' + state`).

Existem várias funções de manipulação de strings que podem ser usadas para outras tarefas, por exemplo:

ASCII(<i>caractere</i>)	retorna o código ASCII de um caractere.
CHAR(<i>inteiro</i>)	retorna o caractere, dado o seu código ASCII
LOWER(<i>expr</i>)	converte para minúsculas
UPPER(<i>expr</i>)	converte para maiúsculas
LTRIM(<i>expr</i>)	retira espaços à esquerda
RTRIM(<i>expr</i>)	retira espaços à direita
REPLICATE(<i>expr</i> , <i>n</i>)	repete uma expressão <i>n</i> vezes
SUBSTRING(<i>expr</i> , <i>início</i> , <i>tamanho</i>)	extrai uma parte de uma string desde <i>início</i> e com <i>tamanho</i> caracteres
RIGHT(<i>expr</i> , <i>n</i>)	retorna <i>n</i> caracteres à direita da string
REVERSE(<i>expr</i>)	inverte uma string
CHARINDEX(<i>'caractere'</i> , <i>expr</i>)	retorna a posição de um caractere dentro da string
SPACE(<i>n</i>)	retorna uma string com <i>n</i> espaços

STR(*número*,*n*,*d*) converte um valor numérico para string, formatado com *n* caracteres na parte inteira (antes da vírgula) e *d* casas decimais depois da vírgula.

STUFF(*expr1*,*início*,*tamanho*,*expr2*) substitui em *expr1*, os caracteres desde *início* até *tamanho* por *expr2*

DATALENGTH(*expr*) retorna a quantidade de caracteres em *expr*

Por exemplo digite o seguinte comando:

```
Select substring(title, 1, 30) Título , str(price, 5, 1) Preço
from titles
```

O que fizemos foi mostrar até 30 caracteres da coluna title e mostramos a coluna price com no máximo 5 números antes da vírgula e 1 casa decimal depois da vírgula.

Então o resultado será:

Título	Preço
The Busy Executive's Database	20.0
Cooking with Computers: Surrep	11.9
You Can Combat Computer Stress	3.0
Straight Talk About Computers	20.0
Silicon Valley Gastronomic Tre	20.0
The Gourmet Microwave	3.0
The Psychology of Computer Coo	NULL
But Is It User Friendly?	22.9

Digite o seguinte comando :

```
Select Replicate('a', 10)
```

Com este comando a letra a foi mostrada 10 vezes.

Funções de Data/hora

O tipo *datetime*, como vimos, armazena datas e horas. Algumas funções trabalham com esse tipo de dados:

DATEADD(*parte*,*número*,*data*) adiciona um certo número de dias (ou meses, anos etc.) à *data*

DATEDIFF(*parte*,*data1*,*data2*) subtrai as duas datas (*data2* - *data1*), retornando um resultado em dias, meses etc. dependendo de *datepart*

DATEPART(*parte*,*data*) retorna a parte especificada da data

DATENAME(*parte*,*data*) retorna o nome por extenso da parte especificada

GETDATE() retorna a data e hora atuais

Nas funções acima, o argumento *parte*, especifica qual parte da data usar. Ele pode ser um dos seguintes valores:

yy o ano

qq o trimestre

mm o mês

Query - dexter.pubs.sa - [untitled] - Select pubdate....*

DB: pubs

```
Select pubdate, datepart(yy, pubdate) Ano from titles
```

pubdate	Ano
1991-06-12 00:00:00.000	1991
1991-06-09 00:00:00.000	1991
1991-06-30 00:00:00.000	1991
1991-06-22 00:00:00.000	1991
1991-06-09 00:00:00.000	1991
1991-06-18 00:00:00.000	1991
1998-11-13 03:10:53.657	1998

Results

Query batch completed. Exec time: 0:00:00 18 rows Ln 1, Col 55

Server 7.0

dy o dia do ano (1-365)

dd o dia do mês

wk o número da semana (0-51)

dw o dia da semana (domingo=1, segunda=2,...)

hh a hora (0-23)

mi os minutos

ss os segundos

ms os milisegundos

Por exemplo, digite o seguinte comando:

```
Select pubdate, datepart(yy, pubdate) Ano from titles
```

O resultado será:

O que fizemos foi mostrar o campo pubdate(Data de publicação do livro) e o de publicação do livro. Para mostrar o ano utilizamos a função de data datepart .

Conversão de dados

A função CONVERT permite converter de um tipo de dado para outro. A sua forma geral de uso é:

CONVERT(*tipo_de_dados, valor*)

Por exemplo:

```
select convert(char(10),au_fname) + ' ' + convert(char(10),au_lname)
from authors
```

Ao executar este comando o resultado será:

Query - dexter.pubs.sa - [untitled] - select convert(...*

DB: pubs

```
select convert(char(10),au_fname) + ' ' + convert(char(10),au_lname)
from authors
```

Abraham	Bennet
Reginald	Blotchet-H
Cheryl	Carson
Michel	DeFrance
Innes	del Castil
Ann	Dull
Marjorie	Green

Results

Query batch completed. Exec time: 0:00:00 23 rows Ln 2, Col 14

Com valores *datetime*, convert pode ter um parâmetro a mais, que especifica o formato de data a ser usado. Os formatos mais usados são 3 (padrão brasileiro dd/mm/aa), 103 (dd/mm/aaaa) e os padrões americanos 1 (mm/dd/yy) e 101 (mm/dd/yyyy). O default é 0, que mostra datas como:

Jan 01 1997 01:13:23 PM

Por exemplo, para ver a data de hoje em formato brasileiro, execute:

```
select convert(char,getdate(),103)
```

Para converter valores numéricos em char pode ser utilizado a função CONVERT, por exemplo:

```
select convert(char,pub_id) from titles
```

Condições de pesquisa

Como vimos, a cláusula WHERE permite selecionar quais as linhas da tabela a serem incluídas no resultado. Existem várias formas de montar uma cláusula WHERE, usando um dos seguintes elementos:

Operadores de comparação:

=	igual a
>	maior que
<	menor que
>= ou !<	maior ou igual (não menor)
<= ou !>	menor ou igual (não maior)
<> ou !=	diferente

<i>Faixas:</i>	BETWEEN <i>valor-ini</i> AND <i>valorl-fin</i>
----------------	--

<i>Listas:</i>	IN (<i>lista</i>)
----------------	---------------------

<i>Casamento de padrões:</i>	LIKE <i>padrão</i>
------------------------------	--------------------

<i>Valores nulos:</i>	IS NULL, IS NOT NULL
-----------------------	----------------------

<i>Combinação de condições:</i>	AND, OR, NOT
---------------------------------	--------------

Usando operadores

As condições mais simples são formadas usando operadores de comparação, como vimos no exemplo anterior:

```
select au_lname, city from authors
where state = 'CA'
```

Note que constantes do tipo *char* (ou *varchar*), bem como datas, devem ser colocadas entre apóstrofos (').

Se quisermos fazer o contrário, isto é, procurar os autores que NÃO são da Califórnia, podemos fazer:

```
select au_lname, city from authors
where state <> 'CA'
```

Note que *diferente* também pode ser representado por !=.

Usando faixas

Na tabela *titles*, para cada livro, está guardada a sua data de publicação na coluna 'pubdate'. Se quisermos saber quais os livros publicados no ano de 1991, podemos fazer a consulta:

```
select pubdate, title
from titles
where pubdate between '1/1/91' and '12/31/91'
```

Onde BETWEEN...AND... seleciona os valores de 'pubdate' que estão dentro de uma determinada faixa. Para fazer o contrário, bastaria usar NOT BETWEEN (mas nesse caso não é tão eficiente a consulta).

Usando listas

Você pode selecionar valores de acordo com uma lista. Se o valor pertence à lista, a linha será incluída no resultado. Por exemplo:

```
select au_lname, city, state
from authors
where state in ('UT','CA')
```

O resultado contém as linhas onde 'state' tem um dos valores 'UT' ou 'CA'. Equivale ao mesmo que usar uma condição composta:

```
where state = 'UT' OR state = 'CA'
```

Para retornar os valores que *não* estão na lista, pode-se usar NOT IN.

Casamento de padrões

O operador LIKE [como] faz casamento de padrões. Um *padrão* é uma string contendo caracteres que podem ser combinados com parte de outra string.

Por exemplo, o caractere % em um padrão representa qualquer quantidade de caracteres. Por exemplo, para obter todos os autores cujo (primeiro) nome começa com A, use:

```
select au_fname, au_lname from authors
where au_fname like 'A%'
```

Para obter todos os nomes que contém as letras 'en' no meio (ou no início ou no fim), use:

```
select au_fname, au_lname from authors
where au_fname like '%en%'
```

Outro caractere para usar em padrões é o sublinhado (_). Ele combina com um único caractere. Por exemplo, se nos seus dados existem pessoas com nome 'Sousa' ou 'Souza', você pode usar: LIKE '%sou_a%'.
 Finalmente, é possível usar os colchetes para combinar com uma determinada faixa de caracteres. Por exemplo, LIKE '[CK]%' encontra os nomes que iniciam com C ou K e LIKE '[A-E]%' os que começam com as letras de A até E. Já LIKE '[^V]%' encontra os nomes que *não* começam com V (o caractere ^ indica não).

Note que as comparações feitas com LIKE dependem da ordem de classificação [*sort order*] escolhida durante a instalação do SQL Server. Se foi usada a ordem "accent-insensitive", como foi recomendado, ele consegue procurar ignorando acentos. Por exemplo, LIKE 'camara' vai encontrar também 'Câmara'.

Procurando valores nulos

O valor NULL no SQL Server indica "não informado" ou "desconhecido". Ele é inserido numa coluna quando aquele valor não é conhecido ou não se aplica. Praticamente qualquer coluna pode ter valores NULL, exceto se tiver sido declarada como NOT NULL na criação da tabela.

NULL não é tratado como outros valores. Especialmente, qualquer operação com NULL tem como resultado NULL e qualquer comparação com NULL tem resultado FALSO. Por exemplo, veja as duas consultas a seguir:

```
select price, title from titles
where price = 19.99
```

```
select price, title from titles
where price <> 19.99
```

Algumas linhas da tabela 'title' têm 'price'=NULL. Essas linhas não vão aparecer nem na primeira nem na segunda consulta, porque NULL=19.99 é falso e NULL<>19.99 também é falso. Para ver essas linhas com valores nulos, use:

```
select price, title from titles
where price is NULL
```

Para ver as linhas onde o valor está preenchido (não NULL), use:

```
select price, title from titles
where price is NOT NULL
```

Usar 'price = NULL' funciona no SQL Server, mas não é compatível com o padrão ANSI.

Juntando várias condições

Você pode fazer condições compostas com AND, OR ou NOT.

O operador AND (E) liga duas condições e retorna verdadeiro apenas se *ambas* são verdadeiras e falso se pelo menos uma delas é falsa. Já OR (OU) retorna verdadeiro se *pelo menos uma* delas for verdadeira e falso se ambas forem falsas. O operador NOT (NÃO) inverte uma condição. Por exemplo:

```
select title, pub_id, price
from titles
where (title like 'T%' OR pub_id = '0877')
AND (price > $16.00)
```

Isso retorna os livros onde:

Ambas as condições 1 e 2 são verdadeiras:

1-**Uma das** seguintes é verdadeira

o título ('title') começa com T,

OU

o código da editora ('pub_id') é '0877'

E

2- o preço ('price') é maior que 16.00

Os parênteses indicam a precedência das condições. Na falta de parênteses, o operador NOT, se presente, é aplicado primeiro, depois as condições com AND são agrupadas, depois as condições com OR são agrupadas. Você pode usar parênteses, mesmo se não necessários, para tornar a expressão mais legível.

Outros recursos

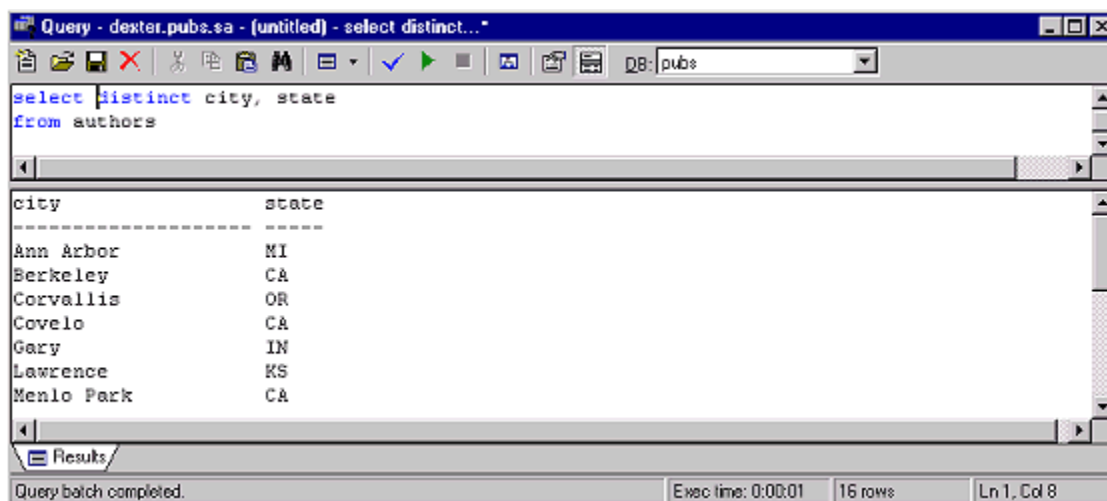
No comando SELECT, a cláusula DISTINCT elimina valores duplicados no resultado e ordena a lista de resultados. Para apenas ordenar por uma ou mais colunas, use ORDER BY. É possível também combinar o resultado de dois comandos SELECT em um único conjunto com o operador UNION.

Eliminando valores duplicados

Se você quiser saber quais as cidades e estados nas quais mora algum autor, pode usar a seguinte consulta:

```
select city, state
from authors
```

Mas note que algumas cidades, como Oakland, CA, aparecem várias vezes. Para eliminar duplicações, use a cláusula DISTINCT. O SQL leva em conta as duas colunas em conjunto, para remover duplicatas.



```
select distinct city, state
from authors
```

O resultado será:

Note que o resultado terá apenas 16 linhas. O SQL Server ordena os dados implicitamente, para poder eliminar as duplicatas.

Ordenando resultados

Para ver o resultado numa ordem particular, use a cláusula ORDER BY. Se estiver presente, deve ser a última cláusula do comando SELECT. Por exemplo, para ver os livros em ordem de preço:

```
select title, type, price
from titles
order by price
```

Você pode indicar após o nome da coluna, se a ordem é ascendente ou descendente, por exemplo:

```
select title, type, price
from titles
order by type asc, price desc
```

Se nem ASC nem DESC estiverem presentes, o default é ASC.

Em vez de colocar o nome da coluna, você pode usar o número relativo (1,2,...). A mesma consulta anterior poderia ser:

```
select title, type, price
from titles
order by 2 asc, 3 desc
```

União de conjuntos

O comando SELECT retorna um *conjunto* de linhas, e permite também operações com a noção matemática de conjuntos. Por exemplo, o resultado de dois comandos SELECT pode ser combinados com o operador UNION. Os dois comandos podem até mesmo trazer dados de tabelas diferentes, desde que com o mesmo número de colunas, e tipos de dados compatíveis para cada coluna correspondente de um com o outro.

Por exemplo, no banco de dados *pubs*, a tabela *authors* contém informação sobre cada autor, o que inclui a cidade e estado onde ele mora (colunas *city* e *state*). A tabela *publishers* contém informação sobre as editoras e suas cidades e estados. Para sabermos o conjunto de todas as cidades onde existem autores ou editoras, pode ser feita uma união dos dois conjuntos, com:

```
select city, state from authors
union
select city, state from publishers
```

Note que na união de dois conjuntos, os elementos repetidos são eliminados, como quando se usa o DISTINCT. O resultado também aparece em ordem crescente, pois o SQL Server ordena os resultados antes de eliminar repetições.

Se você quer ordenar de modo diferente os resultados da união, usando ORDER BY, essa cláusula só pode aparecer no segundo comando SELECT, ou seja, no final dos comandos, por exemplo:

```
select city, state from authors
union
select city, state from publishers
order by state
```

Alteração de Dados

Inserindo Linhas

Atualizando Linhas

Excluindo Linhas

Objetivos:

- Aprender a inserir , atualizar e excluir linhas.

Inserindo linhas

O comando INSERT insere linhas em uma tabela. A forma mais simples do comando INSERT insere somente uma linha , dados os valores.

Sintaxe

```
INSERT [INTO] nome_tabela (colunas)
VALUE (valores)
```

Onde:

nome_tabela é o nome da tabela que deseja incluir os dados.

colunas é o nome das colunas da tabela que deseja acrescentar os dados.

valores é o conteúdo de cada coluna.

Exemplos:

Vamos usar a tabela 'funcionario', do banco de dados Exemplo, criada anteriormente, para inserir linhas. Na janela SQL Query Tool, selecione em "BD" o banco de dados Exemplo. Digite o seguinte:

```
insert into Funcionario
values (1, 'Primeiro Funcionário', 2, 122, 234.23,
'01/01/1998', '01/01/1998', 'M')
```

Nesse caso, são informados os valores de todas as colunas da tabela, na ordem em que elas foram definidas na tabela. Mas é possível também inserir dados parciais de apenas algumas colunas. Para testar, digite os seguintes comandos e depois execute:

```
insert into Funcionario (CodFuncionario, Nome, CodDepartamento, Sexo,
Salario, DataAdmissao)
values (2, 'Segundo Funcionário', 1, 'F', 4360.00, '01/01/1996')
```

```
insert into Funcionario (CodFuncionario, Nome, CodDepartamento, Sexo,
Salario, DataAdmissao, Ramal)
values (3, 'Terceiro Funcionário', 1, 'F', 1500.00, '12/30/1995', 122)
```

```
insert into Funcionario (CodFuncionario, Nome, CodDepartamento, Sexo,
Salario, DataAdmissao)
values (4, 'Quarto Funcionário', 1, 'M', 1500.34, '10/30/1996')
```

```
insert into Funcionario (CodFuncionario, Nome, CodDepartamento, Sexo,
Salario, DataAdmissao)
values (5, 'Quinto Funcionário', 3, 'M', 500.34, '07/30/1997')
```

```
insert into Funcionario (CodFuncionario, Nome, CodDepartamento, Sexo,
Salario, DataAdmissao)
values (6, 'Sexto Funcionário', 3, 'F', 1000.34, '08/30/1995')
```

```
insert into Funcionario (CodFuncionario, Nome, CodDepartamento, Sexo,
Salario, DataAdmissao)
values (7, 'Sétimo Funcionário', 3, 'F', 900.34, '01/01/1997')
```

Nesse caso, os nomes das colunas que serão inseridas são especificados entre parênteses após o nome da tabela. A ordem não precisa ser a mesma das colunas na tabela. Mas a ordem dos valores em VALUES corresponde à ordem dos nomes de colunas informados.

Nesse caso, se uma coluna é omitida da lista, o SQL Server faz o seguinte:

- Se a coluna tem um valor default , o valor default é inserido.
- Caso contrário, se a coluna permite valores NULL, será inserido um NULL.
- Caso a coluna não tenha default e tenha sido criada como NOT NULL, o SQL Server gera uma mensagem de erro e cancela a execução do comando.

Digite agora 'SELECT * from Funcionario'. Você verá que os funcionários 2 , 3 e 4 a data de cadastro é a data de hoje, porque o default para essa coluna é a função GETDATE(). Em outros casos, o valor da coluna ficou NULL quando não informado. A palavra reservada DEFAULT insere o valor default da coluna. Como exemplo execute o seguinte comando:

```
insert into Funcionario
values (8, 'Oitavo Funcionário', 2, 122, 600.23, '01/01/1998', DEFAULT, 'M')
```

Digite agora 'Select * from funcionario where codfuncionario = 8'. Será mostrado os dados do funcionário cujo código é igual a 8. O conteúdo da data do cadastro é a data de hoje, isto ocorreu devido ao seu valor default.

Não acrescente nenhuma linha a mais na tabela de 'funcionario', porque ela será usada posteriormente nos nossos exemplos. Caso acrescente o resultado dos exemplos que iremos utilizar não irá coincidir.

Vamos inserir alguns dados na tabela 'departamento' do banco de dados Exemplo. Esta tabela possui três colunas. A coluna CodDeptSuperior indica o código do departamento que o departamento que esta sendo cadastrado é subordinado.

Execute os seguintes comandos:

```
insert into Departamento Values (1, 'Diretoria', 0)
```

```
insert into Departamento Values (2, 'Departamento Administrativo', 1)
```

```
insert into Departamento Values (3, 'Departamento Pessoal', 1)
```

Não acrescente nenhuma linha a mais na tabela de 'departamento', porque ela será usada posteriormente nos nossos exemplos. Caso acrescente o resultado dos exemplos que iremos utilizar não irá coincidir.

Acrescente dados para a tabela de clientes. Observe que as colunas DataCadastro e País possuem valores Default , as colunas CPF, Cidade, Estado pode conter o valor null.

Usando INSERT com SELECT

Você também pode inserir o resultado de uma consulta SELECT dentro de uma tabela. Para testar, crie uma nova tabela no banco de dados Exemplo, usando o Enterprise Manager (ou com o comando CREATE TABLE), com o nome de CopiaCliente. A tabela deverá ter as seguintes colunas:

Nome	Tipo
Codigo	int
Nome	varchar(50)

Para copiar as linhas 2 e 4 da tabela Cliente, use o seguinte comando:

```
insert into CopiaCliente
    select CodCliente, Nome
    from Cliente
    where CodCliente in (2,4)
```

Cada linha retornada pelo SELECT interno será inserida na tabela CopiaCliente. Esse comando é muito útil para copiar dados entre tabelas semelhantes. Note que nesse caso, as regras que vimos anteriormente ainda se aplicam, a cada linha que o comando está tentando inserir. As colunas da tabela de destino e os valores de resultado do SELECT devem ser compatíveis, ou seja, devem ter o mesmo tipo de dados ou tipos compatíveis e devem estar na mesma ordem (mas os nomes não precisam ser os mesmos, como no caso de 'CodCliente' e 'Codigo').

Se as duas tabelas fossem idênticas, poderia ser usado * no select em vez de uma lista de colunas. Se a tabela CopiaCliente tivesse colunas a mais, além de CodCliente e Nome, teria de ser especificada a lista de colunas a ser inseridos, como vimos anteriormente.

Excluindo linhas

O comando DELETE exclui permanentemente uma ou mais linhas de uma tabela, baseado em alguma condição.

Sintaxe

```
DELETE FROM nome_tabela WHERE condicao
```

Onde:

nome_tabela é o nome da tabela que deseja excluir os dados.

condicao é condição para selecionar as dados que deseja excluir.

Por exemplo, para excluir o cliente nº 2 (Codigo =2) da tabela CopiaCliente, execute o seguinte comando no banco de dados Exemplo:

```
delete from CopiaCliente
    where Codigo = 2
```

Note que a exclusão não pode ser desfeita.

Usando sub-consultas

Assim como UPDATE, o comando DELETE também pode usar sub-consultas para excluir linhas baseado nos dados de outra tabela.

Para testar, copie novamente as linhas de cliente para CopiaCliente e depois insira duas novas linhas em CopiaCliente, como abaixo:

```
insert into CopiaCliente  
  select CodCliente, Nome from Cliente  
insert into CopiaCliente (5, 'Cliente Cinco')  
insert into CopiaCliente (6, 'Cliente Seis')
```

Agora vamos apagar de CopiaCliente apenas as linhas que existem na tabela Cliente.

Para isso, use o comando abaixo:

```
delete from CopiaCliente  
where Codigo in  
  (select CodCliente from Cliente)
```

Limpando uma tabela

Para excluir todas as linhas de uma tabela, existem duas opções. Uma é usar um comando DELETE sem condição WHERE:

```
delete from CopiaCliente
```

Outra opção é o comando TRUNCATE TABLE, que quase sempre é mais rápido que o DELETE, especialmente em tabelas grandes:

```
truncate table CopiaCliente
```

Mas TRUNCATE TABLE não salva informações no log de transações, o que tem algumas consequências com relação a backups, como veremos.

8 - Consultas Avançadas

Dados de Resumo

Junções de Tabelas

Sub-consultas

Objetivos:

- Aprender a gerar dados de resumo com funções agregadas e GROUP BY;
- Aprender a consultar dados a partir de duas ou mais tabelas.

Dados de resumo

Além da sintaxe básica do SELECT que já vimos, alguns elementos a mais podem ser incluídos, as cláusulas GROUP BY e HAVING:

Sintaxe

```
SELECT lista_de_colunas
FROM lista_de_tabelas
WHERE condições
GROUP BY lista_de_expressões
HAVING condições
```

Como já vimos, o WHERE (se presente) separa as linhas que satisfazem as condições iniciais. A cláusula GROUP BY organiza as linhas de resultado em grupos de acordo com os valores das expressões informadas. A cláusula HAVING (opcional) seleciona os grupos de acordo com os resultados. O resultado do SELECT...GROUP BY... tem uma linha para cada grupo, que pode conter *valores de resumo* (somatório, média, contagem etc.) calculados dentro do grupo.

Quando o GROUP BY está presente, é possível usar *funções agregadas*, que calculam valores baseado nas linhas de um grupo e geram valores de resumo.

Exemplo

Para saber quantas linhas existem na tabela '*funcionario*' do banco de dados Exemplo, pode-se usar a função agregada COUNT(*):

```
select count(*)
from funcionario
```

O resultado será 8. Quando a cláusula GROUP BY é omitida, como nesse exemplo, todas as linhas da tabela são agrupadas para formar uma linha de resultado. Você pode também acrescentar uma condição:

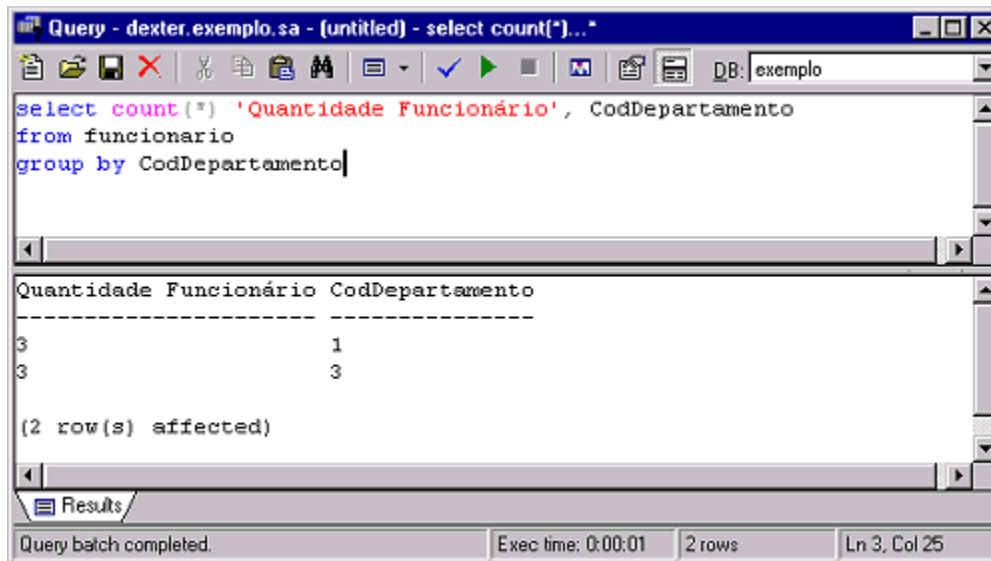
```
select count(*)
from funcionario
where ramal is not null
```

Neste caso o resultado é 3. Estamos contando quantos funcionarios tem a coluna Ramal diferente de null. Quando se utiliza o GROUP BY com o nome de uma coluna,

os resultados são agregados por essa coluna. Por exemplo, para saber quantos funcionarios existem por departamento:

```
select count(*) 'Quantidade Funcionário', CodDepartamento
from funcionario
group by CodDepartamento
```

O resultado será:



Ou seja, todos os três funcionários cujo código do departamento é 1 foram agrupados para gerar uma só linha de resultado. Depois todos os funcionários com coddepartamento = 3 (caso houvesse algum funcionário cadastrado com código departamento = 2, viria logo após o de código departamento = 1) e assim por diante. No resultado é mostrada a coluna 'coddepartamento' (que foi usada no GROUP BY para definir os grupos) e o resultado da função COUNT(*), que é a contagem de elementos de cada grupo.

Funções agregadas

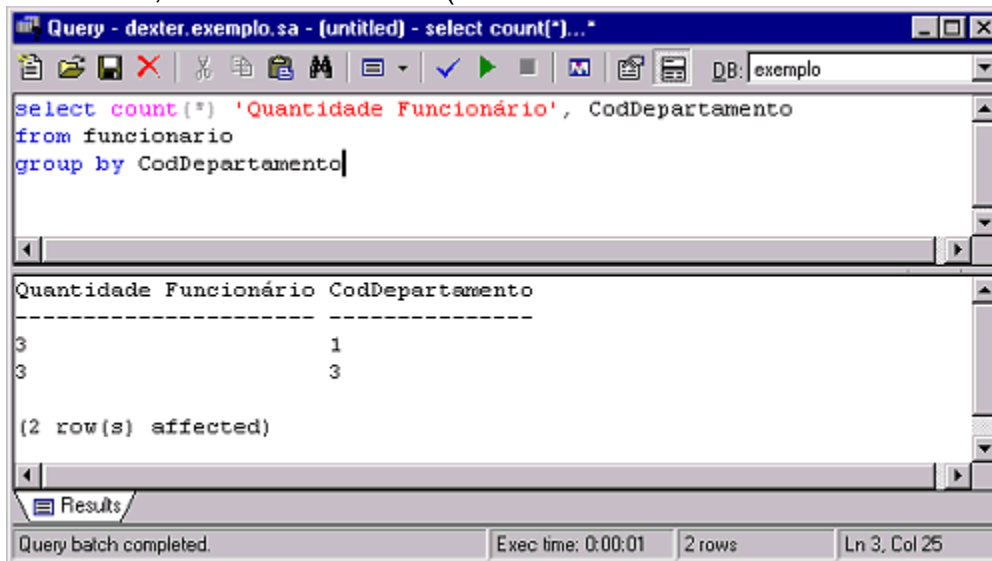
Além da função COUNT, existem outras funções agregadas que podem ser usadas para fazer operações sobre os elementos do grupo:

AVG(<i>expr</i>)	Calcula o valor médio da expressão <i>expr</i> dentro do grupo. A expressão pode ser um nome de coluna ou calculada a partir de colunas e/ou constantes. (Por exemplo, AVG (salario*1.1))
COUNT(<i>expr</i>)	Conta quantos valores existem da expressão dada dentro do grupo (se <i>expr</i> for NULL para uma linha, a linha não é incluída na contagem).
COUNT(*)	Conta quantas linhas existem dentro do grupo.
MAX(<i>expr</i>)	Retorna o máximo valor de <i>expr</i> dentro do grupo.
MIN(<i>expr</i>)	Retorna o mínimo valor de <i>expr</i> dentro do grupo.
SUM(<i>expr</i>)	Retorna o somatório da expressão dentro do grupo.

As funções AVG e SUM podem ser usadas apenas com dados numéricos. As outras podem ser usadas com qualquer tipo de coluna. As funções SUM, AVG e COUNT(*expr*) permitem especificar também o operador DISTINCT, que indica para considerar apenas os valores distintos. Por exemplo, a tabela 'funcionarios' do banco de dados Exemplo contém informação sobre qual departamento este funcionário trabalha. Para saber quais os departamentos que têm ao menos um funcionário, execute o seguinte comando:

```
select count(distinct coddepartamento)
from funcionario
```

O resultado é 3, o número de valores *distintos* de 'departamentos'. Sem usar DISTINCT, o resultado seria 8 (o número de linhas da tabela 'funcionario').



Outro exemplo usando a tabela 'funcionario': para saber o total de salários pagos por cada departamento, pode-se usar GROUP BY e a função SUM calculando o somatório da coluna 'Salario':

```
select coddepartamento Departamento, Sum(Salario) 'Total Salário'
from funcionario
group by coddepartamento
```

O resultado será:

Detalhes do GROUP BY

A cláusula GROUP BY agrupa valores baseado em uma ou mais colunas. No último SELECT acima, group by coddepartamento significa que todas as linhas que têm o mesmo valor da coluna 'coddepartamento' serão agrupadas em uma só. Uma função agregada, como SUM, COUNT, AVG calcula valores sobre todos os elementos do grupo. Uma linha de resumo é gerada para o grupo, contendo o valor representante do grupo, 'coddepartamento' e o resultado de SUM(Salario).

Note que as colunas de resultado da cláusula SELECT (a lista de colunas após o SELECT) podem ser apenas:

- Uma coluna presente na lista do GROUP BY
OU

- Um valor gerado por uma função agregada

Outras colunas não podem ser incluídas no resultado, porque teriam valores diferentes para cada linha do grupo.

Usando a cláusula HAVING

Após feito o agrupamento, pode-se usar a cláusula HAVING para selecionar quais os grupos a serem incluídos no resultado. Por exemplo, para selecionar os departamentos que pagam mais que 1000.00, pode-se fazer:

```
select coddepartamento Departamento, sum(Salario) 'Salário Total'
from funcionario
group by coddepartamento
having sum(Salario) > 1000.00
```

Note que as cláusulas WHERE e HAVING são semelhantes. Mas WHERE seleciona as linhas da tabela que irão participar da geração do resultado. Essas linhas serão agrupadas e *depois* HAVING é aplicado ao resultado de cada grupo, para saber *quais grupos* vão aparecer no resultado. Nas condições usadas por HAVING só podem aparecer valores que sejam os mesmos em todos os elementos do grupo.

Junções de tabelas

Um comando SELECT também pode fazer uma consulta que traz dados de duas ou mais tabelas. Esse é um processo chamado de *junção* [join]. As tabelas têm uma coluna em comum que é usado para fazer as junções.

Antes de executar os exemplos mostrados abaixo, vamos inserir um departamento na tabela 'departamento' que neste caso não terá nenhum funcionário relacionado.

Execute o seguinte comando:

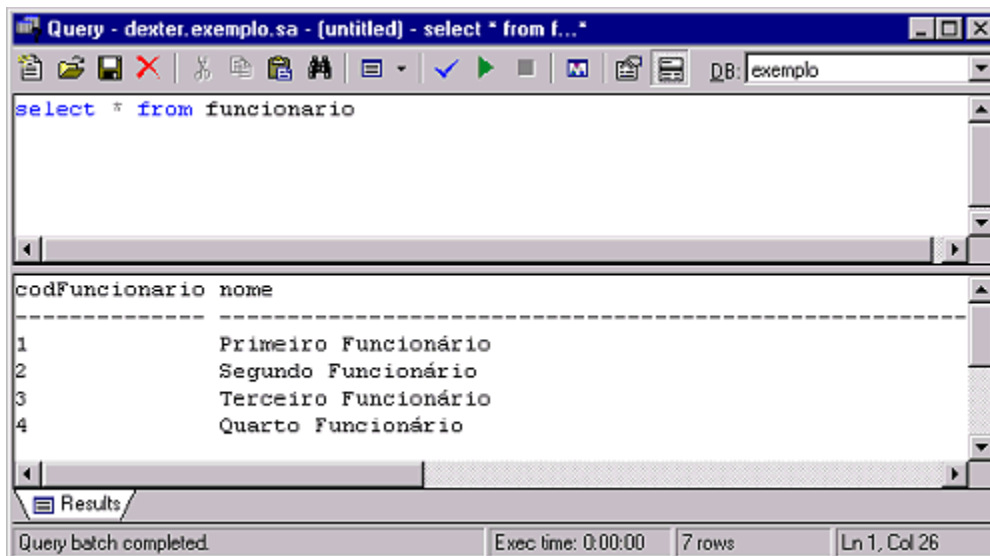
```
insert into departamento values (4, 'Contabilidade', 2)
```

Por exemplo, no banco de dados *Exemplo*, a tabela 'departamento' contém dados de departamentos. Cada departamento tem um número de identificação único, 'CodDepartamento'. Na tabela 'funcionario' estão os dados dos funcionários. Para identificar o departamento do funcionário, a tabela 'funcionario' tem também uma coluna 'CodDepartamento' que pode ser usada para procurar na tabela 'departamento'.

Por exemplo, digite:

```
select * from funcionario
```

O resultado será:



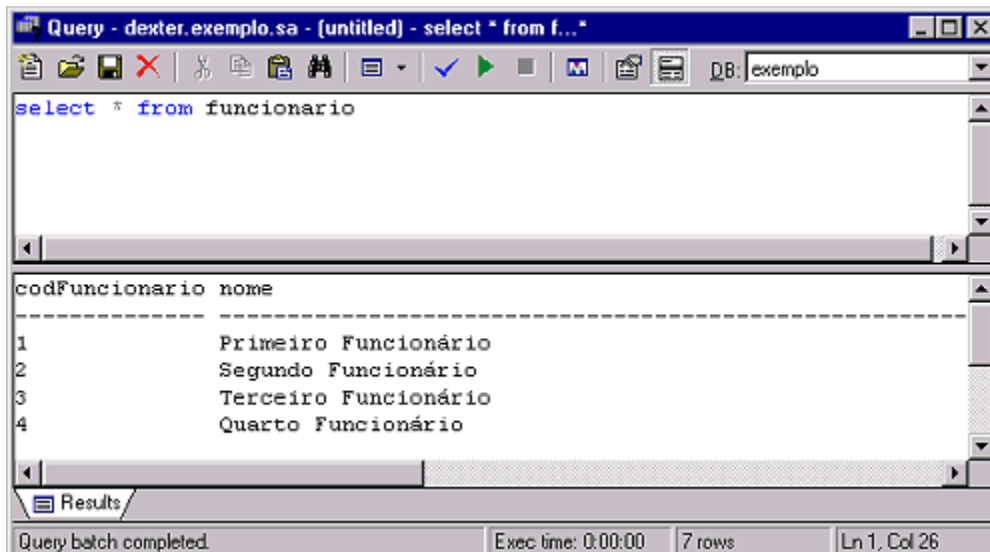
...e veja que na primeira linha da tabela, o nome do funcionário é 'Primeiro Funcionário' e o valor da coluna 'coddepartamento' é 2. Agora digite:

```
select * from departamento
```

Depois de executar, note que na segunda linha o nome do departamento, cujo código é igual a 2, é 'Departamento Administrativo'.

Se quisermos ver uma listagem mostrando os funcionários e seus respectivos departamentos, basta fazer uma junção das duas tabelas. Uma das formas de fazer isso é:

```
select departamento.nome 'Departamento', funcionario.nome
```



```
from funcionario, departamento
```

```
where funcionario.coddepartamento = departamento.coddepartamento
```

O resultado será:

Neste exemplo como as duas tabelas tem o mesmo nome de coluna é necessário qualificar o nome com o nome da tabela, como "funcionario.nome", "departamento.nome". Em geral, é recomendável sempre qualificar os nomes para maior clareza.

Sintaxe da Junção

O SQL Server aceita duas sintaxes diferentes para junção de tabelas. Uma delas, mostrada acima, é específica ao SQL Server e, às vezes, um pouco mais simples de utilizar.

Na sintaxe do SQL Server, na lista do FROM as duas (ou mais) tabelas são especificadas, separadas por vírgulas. Na cláusula WHERE deve haver uma condição ligando as duas, a *condição de junção* [join condition]. Na lista de colunas do SELECT podem ser incluídos colunas de qualquer uma das tabelas.

No exemplo acima foram especificadas as colunas 'departamento.nome' (o nome da departamento, que vem da tabela 'departamento') e 'funcionario.nome' (nome do funcionario, que vem da tabela 'funcionario').

A outra forma de sintaxe que pode ser usada é a sintaxe do padrão ANSI SQL. O exemplo anterior, com a sintaxe ANSI, ficaria:

```
select departamento.nome, funcionario.nome
from funcionario inner join departamento
on funcionario.coddepartamento = departamento.coddepartamento
```

Nessa sintaxe, o *tipo de junção* entre as tabelas deve ser especificado entre elas (veremos os diferentes tipos abaixo) e a condição de junção é especificada com a palavra ON.

Junção interior

O exemplo acima é uma *junção interior* de tabelas [inner join]. Esse tipo de junção conecta as duas tabelas e retorna apenas as linhas que satisfazem a condição de junção. No exemplo, isso significa que, se existirem funcionários para os quais não há departamento relacionado eles não serão incluídos no resultado. Igualmente, se existirem departamentos que não têm empregados (como 'Contabilidade'), eles não aparecem no resultado.

Uma junção interior é chamada de *equijoin* quando as colunas são comparadas usando o =, e as duas colunas aparecem no resultado, mostrando dados redundantes, já que elas têm o mesmo valor. Uma junção interior é chamada *junção natural* quando a coluna usada para junção aparece apenas uma vez no resultado, vinda de uma ou outra tabela.

Na sintaxe ANSI, junções interiores são indicadas da forma:

```
tabela1 INNER JOIN tabela2 ON condição_de_junção
```

Junção cruzada ou irrestrita

Uma *junção cruzada* [cross join] de tabelas, também chamada *junção irrestrita* de duas tabelas gera um resultado formado por *todas* as combinações possíveis de uma linha da primeira tabela com uma linha da segunda. Não existe uma condição de junção. Esse resultado é chamado *produto cartesiano* das duas tabelas. Na sintaxe ANSI, junções cruzadas são indicadas com CROSS JOIN, por exemplo:

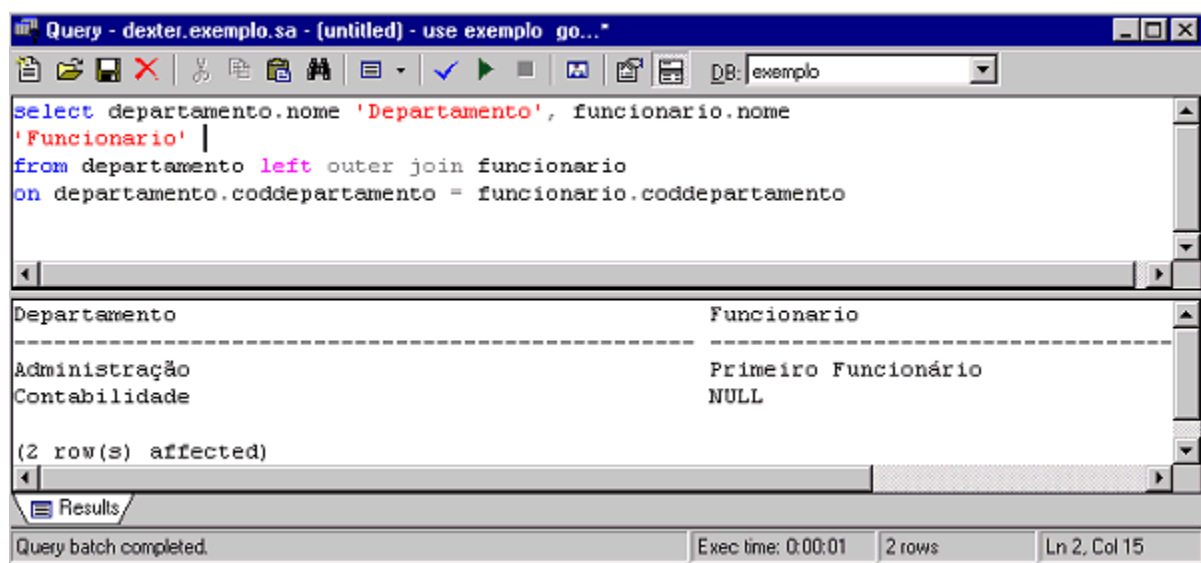
```
select departamento.nome, funcionario.coddepartamento
from funcionario cross join departamento
```

Nesse caso, como a tabela 'funcionario' tem 8 linhas e a tabela 'departamento' tem 4 linhas, o resultado final será 32 linhas que é $8 * 4 = 32$ linhas, formadas por todas as combinações possíveis de funcionarios e departamento.

Na sintaxe do SQL Server, basta especificar a lista de tabelas, sem usar uma condição de junção:

```
select departamento.nome, funcionario.nome, salario
from funcionario, departamento
```

Esse é exatamente um dos problemas com a sintaxe informal: se você incluir mais de uma tabela mas não colocar uma condição de junção (é fácil esquecer uma condição quando existem várias outras envolvidas), o banco de dados vai simplesmente fazer um produto cartesiano sem nenhum aviso.



Junções cruzadas raramente são usadas, mas é importante saber como evitar usá-las. Para isso, sempre crie uma junção interior ou exterior (veremos abaixo), em casos de ter várias tabelas envolvidas.

Junção exterior

Uma junção exterior [outer join] mostra todas as linhas de uma tabela, mesmo quando elas não satisfazem a condição de junção. Por exemplo:

```
select departamento.nome 'Departamento', funcionario.nome
'Funcionario'
from departamento left outer join funcionario
on departamento.coddepartamento = funcionario.coddepartamento
```

O resultado será algo semelhante a isso:

Se é usado LEFT OUTER JOIN indica que todas as linhas da tabela à esquerda (no caso, 'departamento') são incluídas no resultado. Nesse caso, são mostrados todos os departamentos, mesmo aqueles que não tem funcionário. Quando um departamento não tem funcionário, as colunas da tabela 'funcionario' irá mostrar o valor NULL. A

tabela 'departamento' é chamada de a *tabela exterior* e 'funcionario' é a *tabela interior* da junção.

Se fosse usado RIGHT OUTER JOIN, a tabela à direita ('funcionario') mostraria todas as linhas e a tabela à esquerda, apenas as relacionadas. Se for usado FULL OUTER JOIN, todas as linhas de ambas as tabelas são incluídas, mesmo as que não estão relacionadas com a outra tabela.

Junções com mais de duas tabelas

Para falarmos de junções entre duas tabelas crie uma tabela chamada 'cargo', utilize o Enterprise Manager ou o comando Create table. Esta tabela terá as seguintes colunas:

ColumnName	Datatype	Size
CodCargo	int	
Nome	varchar	50

Ao criar esta tabela adicione os seguintes dados:

```
insert into cargo values (1, 'cargo1')
insert into cargo values (2, 'cargo2')
insert into cargo values (3, 'cargo3')
insert into cargo values (4, 'cargo4')
insert into cargo values (5, 'cargo5')
```

Na tabela 'funcionario' acrescente a seguinte coluna:

ColumnName	Datatype
CodCargo	int

Utilize o Enterprise Manager ou o comando Alter Table.

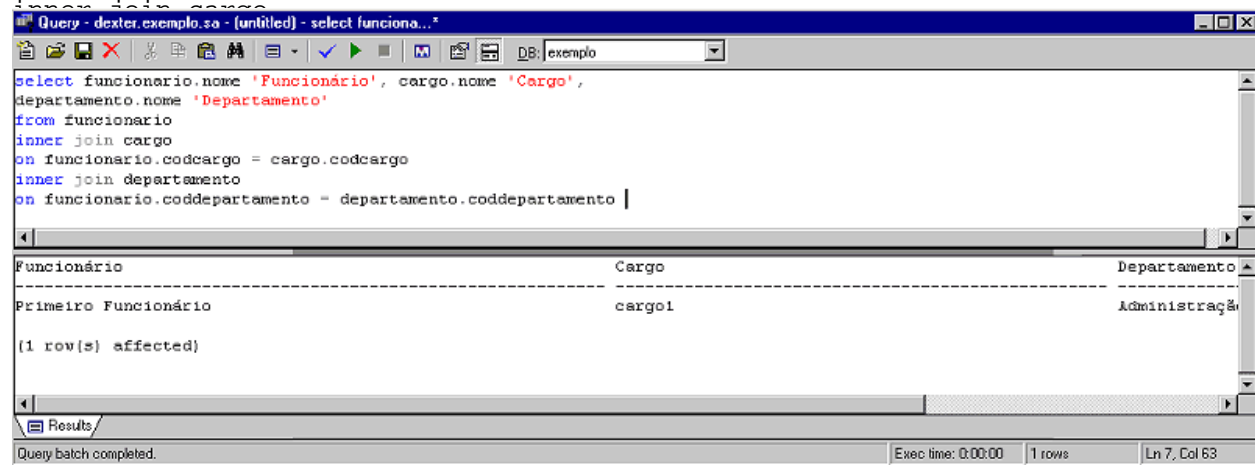
Vamos atualizar a tabela 'funcionario' para acrescentar dados a coluna cargo, execute os seguintes comandos:

```
update funcionario set codcargo = 1 where codfuncionario = 1
update funcionario set codcargo = 2 where codfuncionario = 2
update funcionario set codcargo = 3 where codfuncionario = 3
update funcionario set codcargo = 4 where codfuncionario = 4
update funcionario set codcargo = 5 where codfuncionario = 5
```

É possível juntar três ou mais tabelas com informações relacionadas. No banco de dados 'Exemplos', por exemplo, a tabela 'cargo' contém uma linha que representa cada cargo do funcionário. Entre os dados, esta a identificação do cargo ('codcargo'). A coluna 'codcargo' pode ser usada para buscar o nome do cargo, fazendo uma junção com a tabela 'funcionario'. A tabela 'departamento' contém uma linha que representa cada departamento do funcionário. A coluna 'coddepartamento' pode ser usada para buscar o nome do departamento, fazendo uma junção com a tabela 'departamento'.

Faremos o seguinte:

```
select funcionario.nome 'Funcionário', cargo.nome 'Cargo',
departamento.nome 'Departamento'
from funcionario
inner join cargo
on funcionario.codcargo = cargo.codcargo
inner join departamento
on funcionario.coddepartamento = departamento.coddepartamento |
```



Note que as duas junções de tabela são representadas com um INNER JOIN após o outro. O primeiro INNER JOIN cria uma "tabela" virtual reunindo 'funcionario' e 'cargo'. O segundo reúne essa tabela virtual à tabela 'departamento'.

Apelidos de tabela

Para simplificar a qualificação de colunas, pode-se usar um *apelido* [alias] de tabela, um nome colocado imediatamente após o nome da tabela, na lista do FROM. Esse nome representa a tabela nas qualificações. Por exemplo, a consulta anterior pode ser reescrita da forma:

```
select f.nome 'Funcionário', c.nome 'Cargo' , d.nome 'Departamento'
from funcionario f
inner join cargo c
on f.codcargo = c.codcargo
inner join departamento d
on f.coddepartamento = d.coddepartamento
```

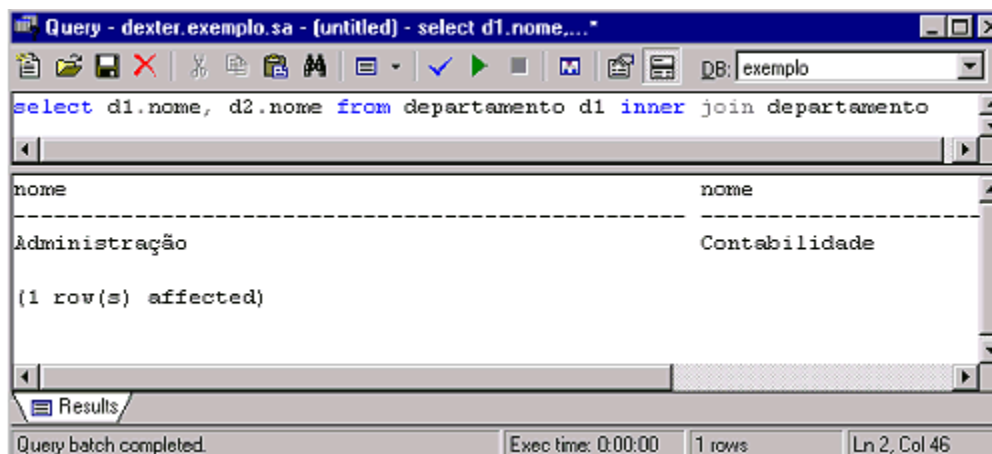
Nesse caso, 'f' é o apelido para a tabela 'funcionario', 'c' para a tabela 'cargo' e 'd' para a tabela 'departamento'. Note que os aliases podem ser usados na lista do SELECT ou nas condições de junção (ou em outros lugares, como numa cláusula WHERE).

Auto-junções

Uma *auto-junção* [self join] é uma junção da tabela com ela mesma. Na tabela departamento, por exemplo, cada departamento está subordinado a outro. A coluna 'coddeptsuperior' indica o código do departamento superior. Para mostrarmos uma lista de todos os departamentos, cada um com seus sub-departamentos, podemos usar:

```
select d1.nome, d2.nome from departamento d1 inner join departamento
d2 on d1.coddepartamento = d2.coddeptsuperior
```

Nesse caso, é obrigatório usar um apelido de tabela para distinguir as duas "cópias" da tabela que estão sendo relacionadas: 'd1' no exemplo representa uma linha da tabela 'departamento' e 'd2' representa outra linha que estão sendo comparadas entre si. O resultado dessa consulta é:



Subconsultas

Uma *sub-consulta* [subquery] é uma consulta SELECT aninhada dentro de outro comando SQL. Ela pode retornar um valor só ou uma lista de valores para ser usada numa comparação.

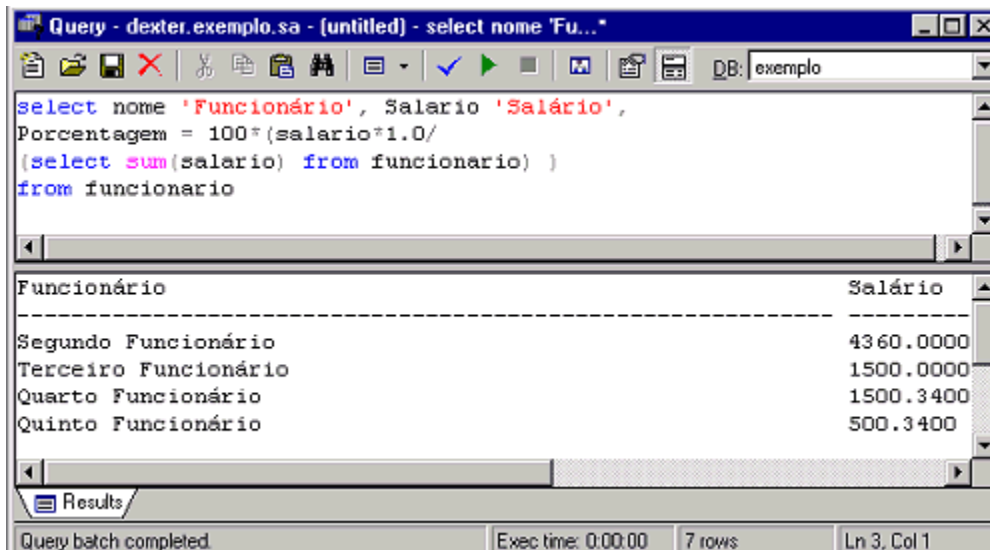
Por exemplo, para saber o gasto que a empresa tem com funcionários, pode-se usar:

```
select sum(salario) from funcionario
```

...que dá como resultado o valor 10.595,82 . Agora suponhamos que queremos saber qual a porcentagem do total que o salário do funcionário representa. Digite o seguinte comando:

```
select nome 'Funcionário', Salario 'Salário',
Porcentagem = 100*(salario*1.0/
(select sum(salario) from funcionario) )
from funcionario
```

O resultado será:



The screenshot shows a SQL Server Query window titled 'Query - dexter.exemplo.sa - (untitled) - select nome 'Fu...'. The query text is:
 select nome 'Funcionário', Salario 'Salário',
 Porcentagem = 100*(salario*1.0/
 (select sum(salario) from funcionario))
 from funcionario
 The results are displayed in a table with two columns: 'Funcionário' and 'Salário'. The 'Porcentagem' column is not visible in the results. The data rows are: 'Segundo Funcionário' (4360.0000), 'Terceiro Funcionário' (1500.0000), 'Quarto Funcionário' (1500.3400), and 'Quinto Funcionário' (500.3400). The status bar at the bottom indicates 'Query batch completed', 'Exec time: 0:00:00', '7 rows', and 'Ln 3, Col 1'.

Funcionário	Salário
Segundo Funcionário	4360.0000
Terceiro Funcionário	1500.0000
Quarto Funcionário	1500.3400
Quinto Funcionário	500.3400

A coluna 'Porcentagem' é calculada como a quantidade dividida pelo resultado da sub-consulta (select sum(salario) from funcionario), que é = 10.595,82 nesse caso. Note portanto que a sub-consulta é executada e seu resultado é substituído dentro da outra consulta, como se tivéssemos escrito:

```
select nome 'Funcionário', Salario 'Salário',
Porcentagem = 100*(salario*1.0/ 10.595,82 )
from funcionario
```

Subconsultas são *sempre* colocadas entre parênteses e podem retornar no resultado apenas uma coluna (embora em alguns casos podem retornar mais de uma linha). No exemplo acima, se a subconsulta retornasse mais de um valor, haveria um erro de execução.

Subconsultas com operadores

Uma sub-consulta pode ser inserida nos resultados, como acima, ou pode ser usada numa expressão WHERE, com um operador de comparação, como =, <, >, <=, >= ou

<>. Nesse caso ela deve retornar apenas um valor. Esse valor é substituído na consulta principal no momento da execução.

Listas de valores

Uma sub-consulta pode retornar uma lista de valores e essa lista de valores pode ser usada em comparações com o operador IN. Por exemplo, para saber quais são os funcionarios que possuem cargos, pode ser usado:

```
select funcionario.nome
from funcionario
where codcargo in (select codcargo from cargo)
```

Uma sub-consulta pode ser usada também com um operador de comparação modificado com as palavras ANY [qualquer] ou ALL [todos]. Por exemplo, > ALL [maior que todos] significa que para que a condição seja satisfeita, o valor comparado deve ser maior que todos os elementos da lista:

O exemplo abaixo utiliza o banco de dados Pubs.

```
select title
from titles
where advance > ALL
(select advance
 from publishers, titles
 where titles.pub_id = publishers.pub_id
 and pub_name = 'Algodata Infosystems')
```

A subconsulta retorna uma lista de valores de 'advance' [adiantamento] que contém todos os valores de adiantamento de livros publicados pela editora 'Algodata Infosystems'. Seu resultado é (5000, 5000, 5000, 7000, 8000) A consulta externa retorna os livros cujo 'advance' é maior do que todos os itens dessa lista.

As seguintes combinações que podem ser usadas:

- > ALL maior que todos os elementos da lista
- < ALL menor que todos
- <> ALL diferente de todos (o mesmo que NOT IN)
- = ANY igual a *algum* dos elementos da lista (o mesmo que IN)
- > ANY maior que algum dos elementos da lista
- < ANY menor que algum dos elementos
- <> ANY diferente de algum dos elementos da lista (falso se igual a todos)

Além disso, podem ser usadas combinações com >= e <=, de forma análoga. Note que não é permitido o uso de = ALL.