Segurança no SQL Server

José Antônio da Cunha

CEFET-RN

Segurança no SQL Server

- Introdução

De modo simplificado, a segurança no acesso às informações significa que o usuário deve ser capaz de acessar os dados necessários com nível de acesso suficiente (e não mais do que suficiente), para que o usuário realize seu trabalho.

Através do mecanismo de segurança também evitamos que pessoas não-autorizadas tenham acesso aos dados.

Segurança no SQL Server

O que será apresentado:

- ✓Uma visão geral da segurança.
- ✓Tipos de segurança disponíveis.
- ✓O papel, criação e administração de Schemas e User Logins.
- ✓Como atribuir permissões aos objetos de um Banco de Dados.
- ✓O que são Roles, como criá-las e administrá-las.
- ✓O planejamento e o gerenciamento da segurança.

Segurança no SQL Server

A segurança no SQL Server 2005 é baseada nos seguintes conceitos:

- ✓Logins.
- ✓User Accounts
- ✓Schemas.
- ✓Roles.
- ✓Permissions.

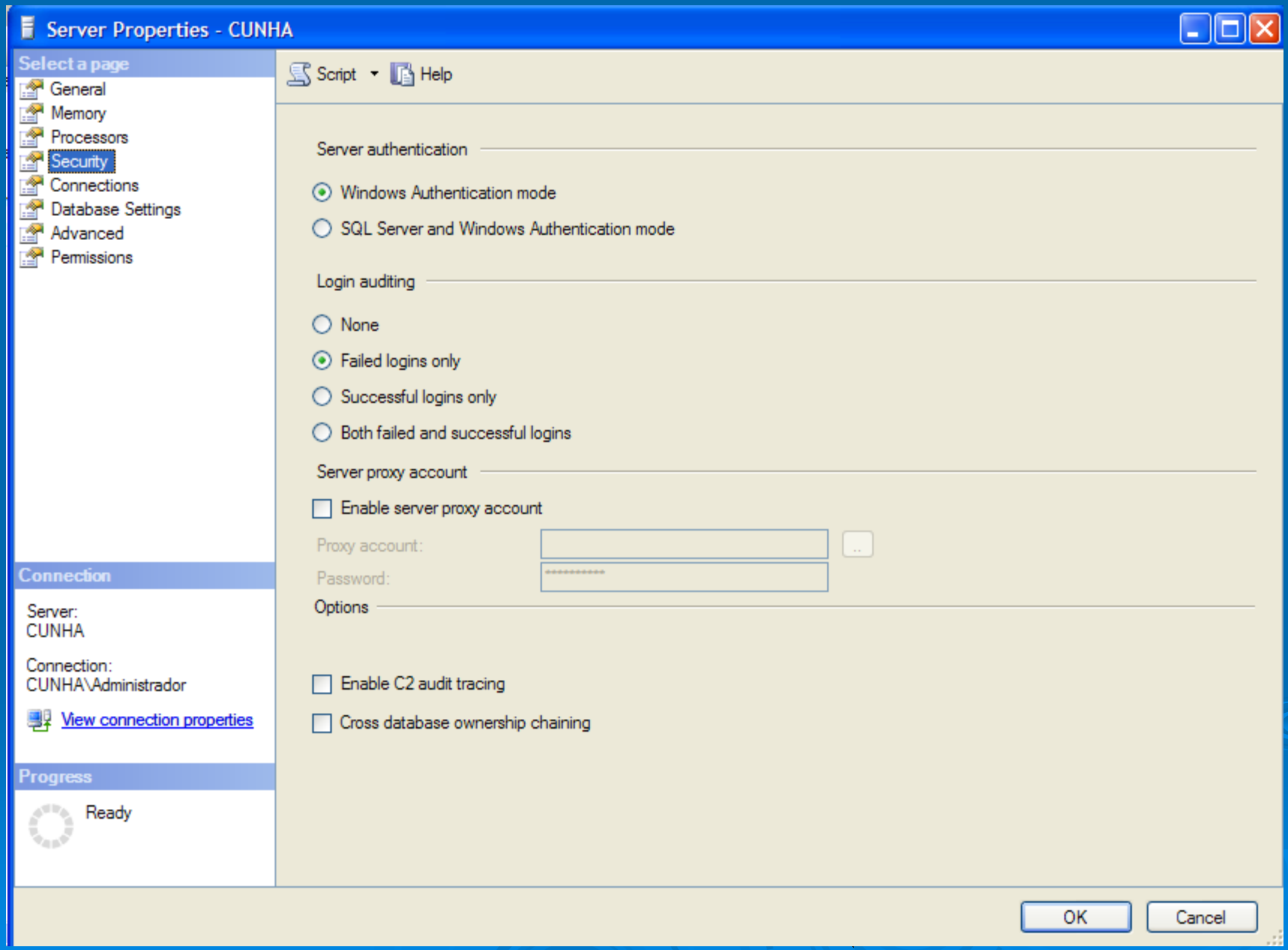
Segurança no SQL Server

Primeiro Precisamos Conectar com o Servidor SQL Server 2005

O primeiro passo para que o usuário possa acessar o servidor SQL Server é estabelecer uma conexão com uma instância do servidor SQL Server.

Os modos de autenticação possíveis são os seguintes:

- ✓**Windows Authentication mode**.
- ✓**SQL Server and Windows Authentication mode**.


Server Properties - CUNHA
Select a page
General
Memory
Processors
Security
Connections
Database Settings
Advanced
Permissions
Script
Help
Server authentication
Windows Authentication mode
SQL Server and Windows Authentication mode
Login auditing
None
Failed logins only
Successful logins only
Both failed and successful logins
Server proxy account
Enable server proxy account
Proxy account:
Password:
Options
Enable C2 audit tracing
Cross database ownership chaining
Connection
Server:
CUNHA
Connection:
CUNHA\Administrador
View connection properties
Progress
Ready
OK
Cancel

Segurança no SQL Server

Permissões para Acessar os Objetos do Banco de Dados

Fazer o Logon no SQL Server não garante acesso a um ou mais Banco de Dados. Você precisa ter permissão de acesso a(os) Banco(s) de Dados e, além do mais, você precisa ter permissão de acesso aos objetos do Banco de Dados.

O usuário precisa passar por dois níveis de segurança:

✓Permissão para fazer a conexão com o SERVIDOR\INSTÂNCIA.

✓Permissão para acessar um ou mais Bancos de Dados.

Conectei com o Servidor, tenho permissão de acesso ao Banco de Dados e ainda não consigo executar uma consulta. O que está acontecendo?

Segurança no SQL Server

O fato de podermos definir permissões para cada objeto nos dá uma grande flexibilidade. A partir desta flexibilidade é que podemos definir diferentes níveis de acesso para diferentes usuários, o que é bastante comum e necessário nas aplicações atuais.

Alguns usuários devem ter permissão de leitura aos dados; outros de leitura e alteração; outros leitura, alteração e exclusão e assim por diante.

Segurança no SQL Server

Schema - Principal

✓**Principal**: um principal é considerado qualquer objeto que possa solicitar acesso a recursos do SQL Server 2005. por exemplo, usuários e grupos do windows, logins e roles do SQL Server, e applications roles.

✓**Schema**: um schema é um conjunto de objetos, sendo que todos os objetos pertencentes a um schema têm como dono o mesmo principal.

No SQL Server 2005 todos os objetos de um Banco de Dados têm como dono um schema.

Servidor.Banco_de_Dados.Schema.Objeto

Observação: No SQL Server 2000 o usuário é dono do objeto e não do schema.

Segurança no SQL Server

No SQL Server 2000 o nome completo fica da seguinte maneira:

Servidor.Banco.Usuário.Objeto

Exemplo:

Cunha.Vendas.jcunha.Cliente

Se fosse necessário, por algum motivo, mudar o dono da tabela cliente para jsilva. Então você teria que mudar o schema para:

Cunha.Vendas.jsilva.Cliente

Segurança no SQL Server

No SQL Server 2005 os objetos do Banco de Dados estão contidos dentro de um schema.

Exemplo:

Cunha.Vendas.Dados.Cliente

Se fosse necessário, por algum motivo, mudar o dono da tabela cliente para jsilva. Então você teria que mudar o schema para:

Cunha.Vendas.Dados.Cliente

Onde: Dados é o schema.

Segurança no SQL Server

Resumindo:

✓Um schema é um container para objetos.

✓Todo objeto pertence a um schema.

✓Todos os objetos de um schema têm o mesmo dono, que é o dono do schema.

✓As permissões podem ser atribuídas para o schema e também para os objetos dentro do schema.

Segurança no SQL Server

Principais vantagens da separação entre usuários e schemas:

✓Múltiplos usuários podem ser donos de um schema, através da definição de uma role ou um grupo do Windows, como dono do schema.

✓O processo de exclusão de um usuário ficou bem mais simplificado. Para excluir um usuário, não é mais necessário alterar o dono de todos os objetos cujo aquele usuário era dono, uma vez que no SQL Server 2005 o usuário não é mais dono do objeto e sim do schema.

✓Com a possibilidade de definição de permissões diretamente em um schema e também nos objetos contidos no schema, podemos definir um nível de permissões muito mais granular do que no SQL Server 2000.

Segurança no SQL Server

Criando Logins com Comandos T-SQL

Tabela 1 Comandos para adicionar logins

Comando	**Utilizado**
Sp_grantlogin	Para adicionar logins do Windows 2000. podemos adicionar usuários ou grupos. Utilizamos o formato DOMINIO\nome ou SERVIDOR\nome.
Sp_addlogin	Para adicionar novos logins do SQL Server, para o caso de estarmos utilizando o modo de segurança SQL Server e Windows Authentication mode.

Segurança no SQL Server

Criando Logins com Comandos T-SQL

Sintaxe para o comando sp_grantlogin:

exec sp_grantlogin 'DOMINIO\nome'

Ou

exec sp_grantlogin [DOMINIO\nome]

Ou

exec sp_grantlogin 'SERVIDOR\nome'

Por exemplo, para adicionar o usuário chico, do domínio CUNHA, utilize o seguinte comando:

```
Exec sp_grantlogin 'CUNHA\chico'
```

Segurança no SQL Server

Podemos remover a permissão de login para um usuário ou grupo do Windows, utilizando o comando **sp_revokelogin**.

Sintaxe para o comando sp_revokelogin:

exec sp_revokelogin 'DOMINIO\nome'

Ou

exec sp_revokelogin [DOMINIO\nome]

Ou

exec sp_revokelogin 'SERVIDOR\nome'

Por exemplo, para remover a permissão de login do usuário chico, do domínio CUNHA, utilize o seguinte comando:

```
Exec sp_revokelogin 'CUNHA\chico'
```

Segurança no SQL Server

Observação sobre o comando sp_revokelogin:

✓Ao removermos a permissão de login, o usuário não poderá mais conectar-se com o servidor SQL, a menos que um dos grupos aos quais o usuário pertença tenha permissão de login. Lembre-se que o usuário sempre herda as permissões do grupo.

Podemos negar, explicitamente, a permissão de login para um usuário ou grupo do Windows, utilizando o comando sp_denylogin. Neste caso, a conta do usuário ou grupo continua na lista de logins, porém com o direito de conexão com o servidor SQL, explicitamente negado.

Segurança no SQL Server

Sintaxe para o comando sp_denylogin:

Exec sp_denylogin ‘DOMINIO\nome’

Ou

Exec sp_denylogin [DOMINIO\nome]

Ou

Exec sp_denylogin ‘SERVIDOR\nome’

Por exemplo, para negarmos, explicitamente, a permissão de login do usuário chico, do domínio CUNHA, utilize o seguinte comando:

```
Exec sp_denylogin ‘CUNHA\chico’
```

Segurança no SQL Server

Observações sobre o comando sp_denylogin:

✓Sp_denylogin não pode ser executado como parte de uma transação definida pelo usuário ou por um aplicativo que o usuário está utilizando.

✓Para permitir que o usuário volte a se conectar, removendo o efeito de sp_denylogin, podemos utilizar sp_grantlogin.

Segurança no SQL Server

Criando Logins com Comandos T-SQL

Agora vamos tratar dos comandos para adicionar e remover logins do próprio SQL Server,os quais podem ser utilizados, quando a intância do SQL Server estiver configurado para o modo de autenticação SQL Server and Windows Authentication.

Para isso utilize o comando **sp_addlogin.**

Segurança no SQL Server

Sintaxe do comando sp_addlogin:

```
Sp_addlogin [ @loginame = ] 'login'
        [ , [ @passwd = ] 'password' ]
        [ , [ @defdb = ] 'database' ]
        [ , [@deflanguage = ] 'language' ]
        [ , [ @sid = ] sid ]
        [ , [ @encryptopt = ] 'encryption_option' ]
```

Observe que podemos definir uma série de opções, tais como a senha,o Banco de Dados associado ao login, um identificador de segurança único (sid) e a definição se a senha vai ser criptografada ou não ao ser armazenada no servidor.

Sid é um varbinary(16)

Segurança no SQL Server

Por exemplo, para adicionarmos o usuário flavio1, com senha em branco e associado ao Banco de Dados Northwind

```
Exec sp_addlogin 'flavio1', '', 'Northwind'
```

Para excluir um login SQL Server, podemos utilizar o comando sp_droplogin.

Sintaxe para o comando sp_droplogin:

Exec sp_droplogin 'nome'

Por exemplo, para excluir o usuário flavio1, podemos utilizar o seguinte comando:

```
Exec sp_droplogin 'flavio1'
```

Segurança no SQL Server

Algumas observações sobre o comando sp_droplogin:

- ✓Sysadmin e securityadmin têm permissão para utilizar esse comando.
- ✓Caso o login que está sendo excluído esteja adicionado como usuário de algum Banco de Dados, o login não poderá ser excluído. Você tem que excluí-lo primeiro do Banco de Dados.
- ✓Os seguintes logins não poderão ser excluídos:
  - ✓O login de administração sa
  - ✓Um login que esteja atualmente conectado com o servidor.

Segurança no SQL Server

Ao criarmos um login no SQL Server, devemos levar os seguintes fatos em consideração:

1. Um login não pode conter o caractere de barra invertida \ como parte do nome.
2. Logins e senhas podem conter até 128 caracteres, incluindo letras, símbolos e dígitos.
3. Não podemos adicionar um login com o mesmo nome de um login reservado, como por exemplo sa ou public.
4. O nome de login não pode conter o valor NULL ou ser uma string vazia ‘’.

Segurança no SQL Server

Criando Roles

Podemos utilizar **Roles** para simplificar a atribuição de permissões de acesso aos objetos do SQL Server. Os Roles são semelhantes ao conceito de grupos de usuários do Windows.

Por exemplo – **Role → FinançaConsulta** e outra **Role → FinançaAlteração**

No Banco de Dados **Finanças** damos permissões somente de **leitura** para a role FinançaConsulta e de **leitura/Escrita** e **exclusão** para a role FinançaAlteração. Depois, incluímos os usuários que precisam de acesso somente leitura na role FinançaConsulta, e os que precisam de acesso de manutenção no banco, incluímos na role FinançaAlteração.

Se um usuário não deve mais ter acesso de alteração, é só retirá-lo da role FinançaAlteração.

Segurança no SQL Server

Criando Roles

Existem algumas roles que já são criadas no momento da instalação. Temos as chamadas Server Roles e as Databases Roles

Tabela 2 Permissões associadas com as principais Server Roles.

Role	Permissões para os membros dessa Role
Sysadmin	Poderes totais sobre todos os objetos do servidor.
Securityadmin	Pode gerenciar logins do servidor.
Serveradmin	Pode configurar a maioria das opções do servidor.
Diskadmin	Gerenciar os arquivos de um banco de dados.
Dbcreator	Criar e alterar Banco de Dados.
Processadmin	Gerenciar processos rodando no SQL Server.
Setupadmin	Pode gerenciar e configurar a replicação entre servidores SQL Server e extender store procedures.

Segurança no SQL Server

Observação: não é possível criar novas roles de servidor

Também temos algumas roles predefinidas para Banco de Dados. Na Tabela 3, temos a descrição destas roles.

Segurança no SQL Server

Tabela 3 Permissões associadas com as principais Fixed Databases Roles.

Role	**Permissões para os membros desta role.**
db_owner	Tem poderes totais sobre o banco de dados.
db_accessadmin	Pode adicionar e remover usuários ao Banco de Dados.
db_datareader	Pode ler dados em todas as tabelas de usuário do BD.
db_datawriter	Pode adicionar, alterar ou excluir dados em todas as tabelas de usuário do BD.
db_ddladmin	Pode adicionar, modificar ou excluir objetos do BD.
db_securityadmin	Pode gerenciar roles e adicionar ou excluir usuários às roles do BD. Pode gerenciar as permissões para objetos do BD.
db_backupoperator	Pode fazer o backup do BD.
db_denydatareader	Não pode consultar dados em nenhuma das tabelas do BD, mas pode efetuar alterações na estrutura do BD e de seus objetos.
db_denydatawriter	Não pode alterar dados no Banco de Dados.

Segurança no SQL Server

Criando Roles

Existe ainda uma role chamada **public**. Todos os usuários adicionados a um Banco de dados automaticamente pertence a esta role. Não podemos adicionar novos usuários a esta role, pois qualquer usuário ou role que é adicionado ao Banco de Dados fará parte desta role.

A **Role Public** não pode ser excluída.

Devemos ter cuidado com as permissões atribuídas a ela, uma vez que todos os usuários do Banco de Dados fazem parte dela.

Se a role public for corretamente utilizada, pode simplificar a administração, em determinadas situações.

Segurança no SQL Server

Criando Roles

Por exemplo, se todos os usuários de um Banco de Dados devem ter acesso de leitura, basta dar permissão de leitura para a role public. Como todos os usuários pertencem à role public, eles irão herdar a permissão de leitura, atribuída à role public.

Segurança no SQL Server

Criando Novas Roles Usando T-SQL

Para adicionar uma nova role a um Banco de Dados, utilizamos o comando **sp_addrole**.

Sintaxe:

exec sp_addrole ‘nome’, ‘dono da role’

Por exemplo, para adicionar uma role chamada teste, cujo dono seja chamado Cliente, faça:

exec sp_addrole ‘teste’, ‘Cliente’

Notas:

•Se não for especificado o parâmetro dono, o dono da role será o Schema dbo – database owner.

•Não podemos criar novas roles à nível de servidor, somente à nível de BD.

Segurança no SQL Server

Criando Roles

Para excluir uma role, podemos utilizar o comando sp_droprole.

Sintaxe:

exec sp_droprole 'nome'

Exemplo:

exec sp_droprole 'teste'

O comando sp_helprole – fornece informações sobre todas as roles do BD.

Exemplo:

Use Northwind

exec sp_helprole

Segurança no SQL Server

Dando Permissão de Acesso ao Banco de Dados (T-SQL)

Para adicionar um logins à lista de usuários autorizados a acessar um Banco de Dados, utilizamos o comando **sp_grantdbaccess**.

Sintaxe:

use database

exec sp_grantdbaccess ‘nome de login’

Por exemplo, para adicionar o usuáro user1, do domínio CUNHA ao Banco de Dados Northwind, faça:

Use Northwind

Exec sp_grantdbaccess ‘CUNHA\user1’

Segurança no SQL Server

Revogando acesso

Para retirar a permissão de acesso do usuário a um Banco de Dados, utilize o comando **sp_revokedbaccess**.

Por exemplo, para remover o usuário CUNHA\user1, do Banco de Dados Northwind, faça:

Use Northwind

Exec sp_revokedbaccess 'CUNHA\user1'

Segurança no SQL Server

Adicionando Usuários Como Membro de Uma ou Mais Roles

Para adicionar um usuário a uma role de servidor, utilize o seguinte comando:

Exec sp_addsvrrolemember ‘login’, ‘role’

Por exemplo, para adicionar os usuários user 1 e user2 do servidor CUNHA à role sysadmin, faça:

Exec sp_addsvrrolemember ‘CUNHA\user1’, ‘sysadmin’

Exec sp_addsvrrolemember ‘CUNHA\user2’, ‘sysadmin’

Segurança no SQL Server

Para excluir um usuário de uma role do servidor, utilize o comando **sp_dropsvrrolemember**.

Por exemplo, para excluir os usuários user1 e user2 do servidor CUNHA da role sysadmin, faça:

Exec sp_dropsvrrolemember 'CUNHA\user1', 'sysadmin'

Exec sp_dropsvrrolemember 'CUNHA\user2', 'sysadmin'

Segurança no SQL Server

Para adicionar um usuário a uma role de Banco de Dados, utilize o seguinte comando:

Use database

Exec sp_addrolemember ‘role’, ‘usuário’

Por exemplo, para adicionar os usuários user1 e user2 do servidor CUNHA, como membros da role consulta, do Banco de Dados Vendas, faça:

Use Vendas

Exec sp_addrolemember ‘Consulta’, ‘CUNHA\user1’

Exec sp_addrolemember ‘Consulta’, ‘CUNHA\user2’

Segurança no SQL Server

Para excluir um usuário de uma role de Banco de Dados, utilize o seguinte comando **sp_droprolemember**.

Por exemplo, para excluir os usuários user1 e user2 do Servidor CUNHA, da role Consulta, do Banco de Dados Vendas, faça:

Use Vendas

Exec sp_droprolemember ‘Consulta’, ‘CUNHA\user1’

Exec sp_droprolemember ‘Consulta’, ‘CUNHA\user2’

Segurança no SQL Server

Dando Permissões Para Banco De Dados

Para um Banco de Dados, podemos definir, dentre outras, as seguintes permissões:

- ✓**Create Table.**
- ✓**Create View.**
- ✓**Create SP.**
- ✓**Create Rule.**
- ✓**Create Function.**
- ✓**Backup DB.**
- ✓**Backup Log**
- ✓**Etc.**

Segurança no SQL Server

Para atribuir permissões com a T-SQL, utilize o comando **GRANT**.

Sintaxe:

GRANT { ALL | statement [,...n] } TO security_account [,...n]

Exemplo1: Garantir para o login CUNHA\user1 a permissão de criar novos Bancos de Dados:

GRANT CREATE DATABASE TO [CUNHA\user1]

Login de instância Windows

Nota: O Banco de Dados atual deve ser o Master.

Segurança no SQL Server

Exemplo2: Atribuir as permissões CREATE TABLE, CREATE RULE e CREATE VIEW, para o usuário user1 do servidor CUNHA no Banco de Dados Northwind.

Use Northwind

GRANT CREATE TABLE, CREATE RULE, CREATE VIEW

TO [CUNHA\user1]

Exemplo3: Atribuir as permissões CREATE TABLE, CREATE RULE e CREATE VIEW, para o usuário user1e user2 do servidor CUNHA no Banco de Dados Northwind.

Use Northwind
GRANT CREATE TABLE, CREATE RULE, CREATE VIEW
TO [CUNHA\user1], [CUNHA\user2]

Segurança no SQL Server

Exemplo4: Atribuir todas as permissões para os usuários user1 e user2 do servidor CUNHA, no Banco de Dados Northwind.

Use Northwind

GRANT ALL TO [CUNHA\user1], [CUNHA\user2]

Para retirar as permissões de Banco de Dados, utilize o comando **REVOKE**.

Sintaxe:

REVOKE { ALL | statement [,...n] } FROM security_account [,...n]

Segurança no SQL Server

Exemplo1: Retirar a permissão de criar novos Bancos de Dados, atribuída para o login CUNHA\user1, anteriormente.

REVOKE CREATE DATABASE TO [CUNHA\user1]

Nota: O Banco de Dados Master deve ser o atual.

Exemplo2: Retirar todas as permissões atribuídas ao usuário user1 do servidor CUNHA, no Banco de Dados Northwind.

Use Northwind

REVOKE ALL TO [CUNHA\user1]

Segurança no SQL Server

Dando Permissões A Objetos Do Banco De Dados

Principais permissões de objetos de Banco de Dados:

- ✓SELECT.
- ✓INSERT.
- ✓DELETE.
- ✓UPDATE.
- ✓EXECUTE.
- ✓REFERENCES.

Segurança no SQL Server

Para atribuir permissões de objetos do Banco de Dados utilize o comando **GRANT**.

Exemplo1: Garantir para o usuário user1 de CUNHA a permissão de selecionar novos registros e atualizar os registros existentes, na tabela Cliente do Banco de Dados vendas.

```
Use Vendas

GRANT SELECT, UPDATE ON Cliente TO [CUNHA\user1]
```

Segurança no SQL Server

Exemplo2: Garantir para o usuário user1 e user2 de CUNHA a permissão de selecionar novos registros, atualizá-los e excluí-los, na tabela Cliente do Banco de Dados vendas.

```
Use Vendas

GRANT SELECT, UPDATE, DELETE ON Cliente

TO [CUNHA\user1], [CUNHA\user2]
```

Para retirar as permissões de objetos do Banco de Dados utilize o comando **REVOKE**.

Segurança no SQL Server

Exemplo1: Retirar a permissão UPDATE, atribuída para o usuário user1 do servidor CUNHA, anteriormente.

```
Use Vendas

REVOKE UPDATE ON Cliente TO [CUNHA\user1]
```

Exemplo2: Retirar todas as permissões atribuídas ao usuário user2, na tabela Cliente do Banco de Dados Vendas.

```
Use Vendas

REVOKE ALL ON Cliente TO [CUNHA\user2]
```

Segurança no SQL Server

Para negar as permissões de objetos do Banco de Dados utilize o comando **DENY**.

Eemplo1: Negar permissão UPDATE, para o usuário user1 do servidor CUNHA, na tabela Cliente, do Banco de Dados Vendas.

```
Use Vendas

DENY UPDATE ON Cliente TO [CUNHA\user1]
```

Eemplo2: Negar permissão SELECT, UPDATE e DELETE, para os usuário user1 e user2 do servidor CUNHA, na tabela Cliente, do Banco de Dados Vendas.

```
Use Vendas

DENY SELECT, UPDATE, DELETE ON Cliente

TO [CUNHA\user1], [CUNHA\user2]
```

Segurança no SQL Server

Trabalhando com Schema

Sem dúvida umas das principais mudanças que ocorreu entre as versões anteriores e o SQL Server 2005 foi a separação feita entre usuários e Schemas. Onde não existe mais o conceito de dono dos objetos de um Banco de Dados, tais como tabelas, views e stored procedures. No SQL Server 2005, todos os objetos pertencem a um schema e temos o dono do schema e não mais o dono dos objetos.

Segurança no SQL Server

Trabalhando com Schema

Vamos aprender a executar as seguintes tarefas:

- ✓Criar novos schemas.
- ✓Atribuir objetos a schema.
- ✓Alterar o dono de um schema.

Segurança no SQL Server

Trabalhando com Schema

Exemplo1: Vamos criar um schema chamado Producao, dentro do Banco de Dados Empresa, da instância CUNHA. Para isso, siga os passos a seguir:

1. Abra o SQL Server Management Studio.
2. Na janela Object Explorer, navegue até o Banco de Dados Empresa.
3. Clique no sinal de + ao lado do Banco de Dados Empresa.
4. Clique no sinal de + ao lado de Security.
5. Clique no sinal de + ao lado de Schema e observe a lista de schemas já definidos.
6. Clique com o botão direito do mouse na opção Schema e selcione a opção New Schema.

Segurança no SQL Server

Trabalhando com Schema

7. No campo Name digite Producao e, para o dono deste schema, vamos especificar a role Gerentes. No campo Schema Owner digite Gerentes.
8. Clique OK. O novo schema está criado.

Ao criar um schema você poderá criar novos objetos e adicioná-los a este schema, poderá alterar as propriedades dos objetos já existentes, para que passem a fazer parte deste schema e poderá atribuir permissões de acesso, diretamente ao schema.

Segurança no SQL Server

Trabalhando com Schema

O SQL Server atribuirá o novo objeto que está sendo criado, ao schema definido como Default schema (dbo), para o usuário logado.

Como exemplo vamos criar uma nova tabela (Venda) no Banco de Dados Empresa e associar esta tabela ao schema Producao.

1. Selecione a opção New Table.
2. Abra a janela de propriedades (F4).
3. Selecione Schema → Producao

Schema

Producao.Vendas

Schema

O default seria → dbo.Vendas

Segurança no SQL Server

Trabalhando com Schema

Para alterar o dono de um schema, siga os seguintes passos:

1. Abra o SQL Server Management Studio.
2. Localize o schema a ser alterado.
3. Dê um clique duplo no schema, para abrir a janela de propriedades do schema.
4. Na janela propriedades, na Guia Geral, no campo schema Owner, basta digitar o nome do novo dono (que pode ser um usuário ou uma role).
5. Clique OK.