

Curso de Especialização:  
Engenharia e Administração de Sistemas de Banco de Dados

# Fundamentos de Sistemas de Banco de Dados



## Transact SQL – DCL

Profa. Dra. Gisele Busichia Baioco

[gisele@ft.unicamp.br](mailto:gisele@ft.unicamp.br)



UNICAMP



FACULDADE DE TECNOLOGIA



# Conteúdo

---

- Controle de Acesso: *Logins*, Usuários (*Users*), *Roles*, *Schemas* e Permissões;
- *Backup/Restore*;
- Agendamento de Tarefas: *Jobs*.



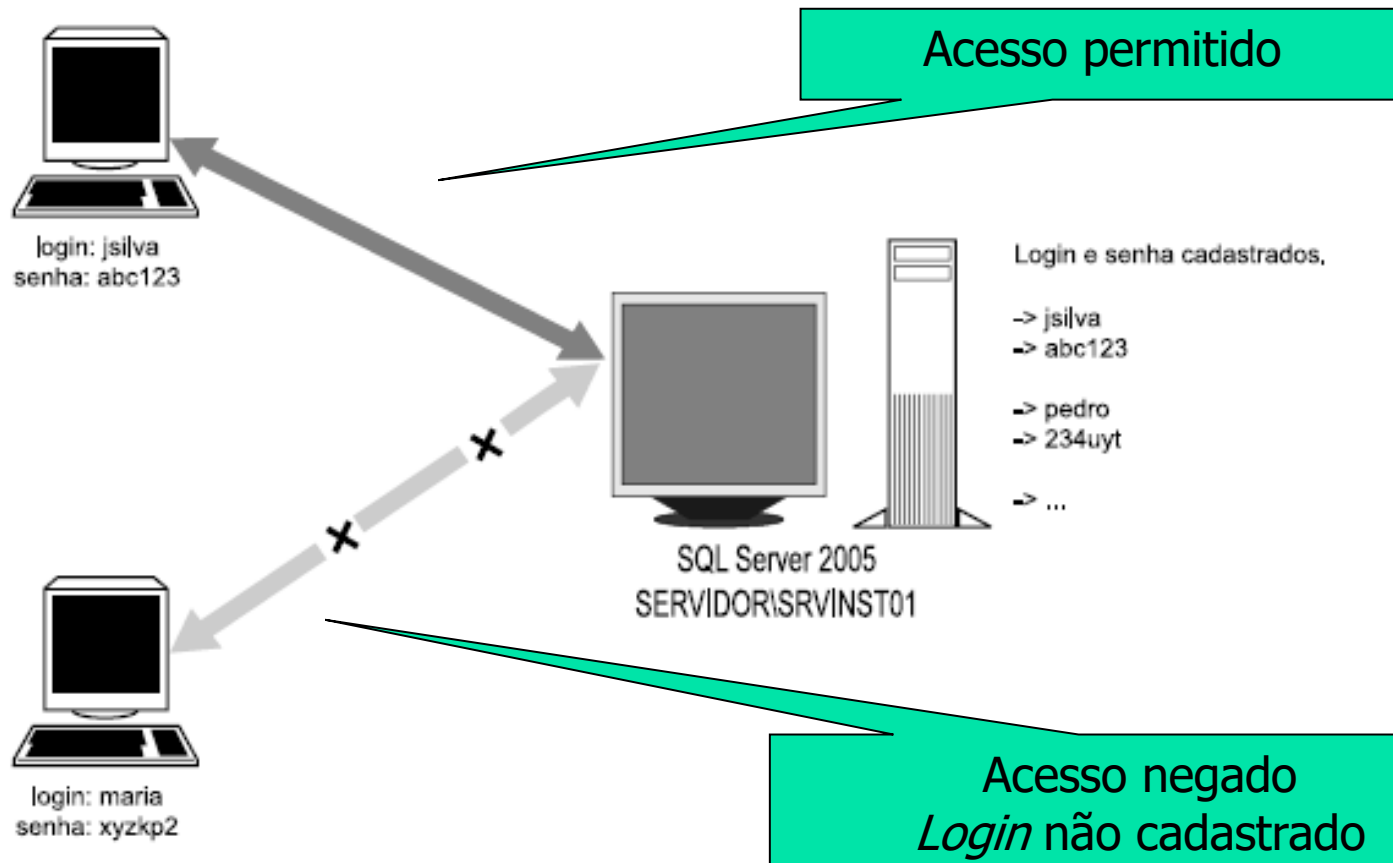
# Controle de Acesso

---

- Três níveis de acesso são necessários para que um usuário consiga acessar dados de um BD:
  1. Acesso (conexão) ao servidor SQL Server – *Logins*;
  2. Acesso aos BDs de um servidor SQL Server – *Users*;
  3. Acesso aos objetos dos BDs do servidor SQL Server – Permissões.

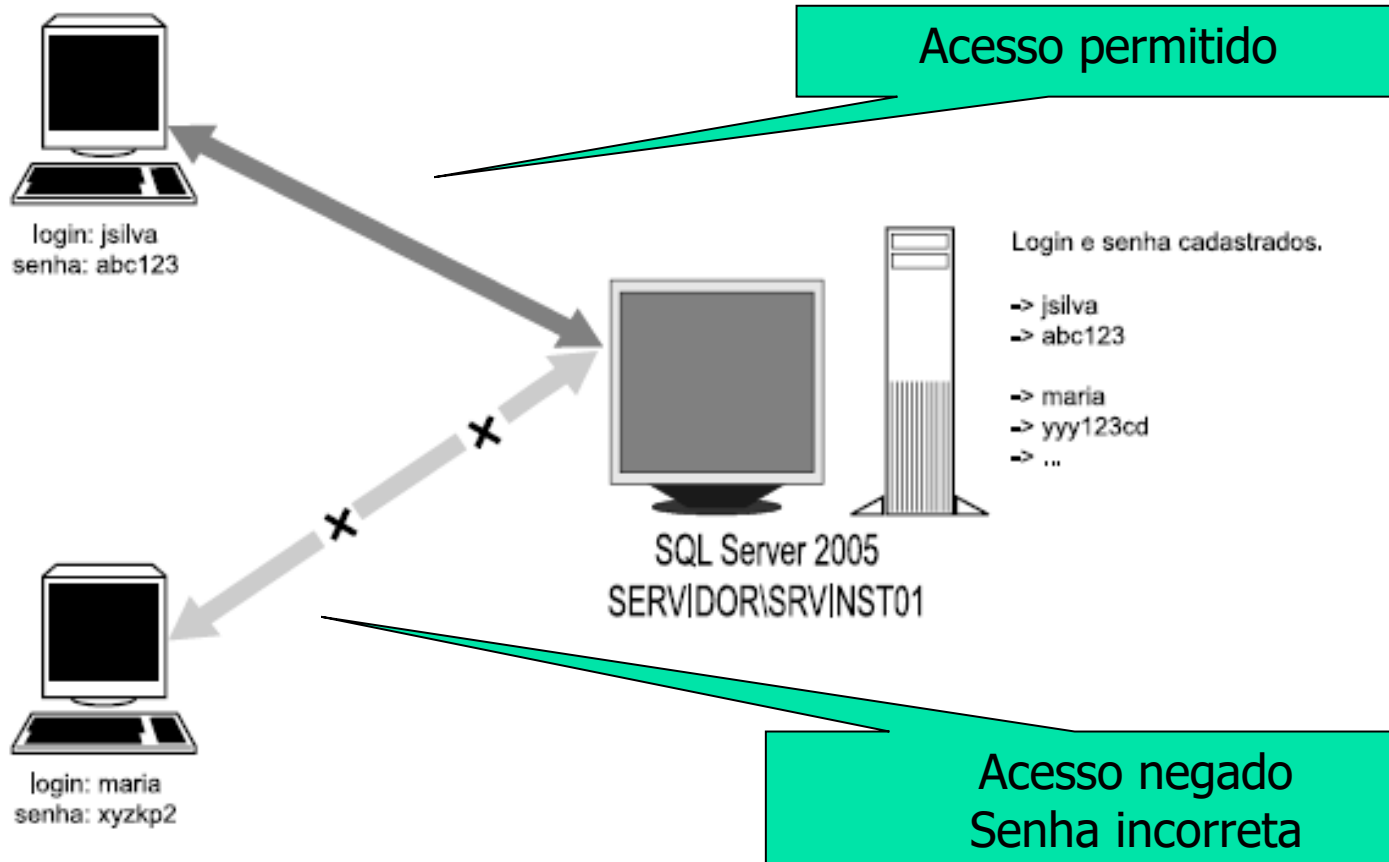
# Controle de Acesso

## Acesso ao Servidor – *Logins*



# Controle de Acesso

## Acesso ao Servidor – *Logins*





# Controle de Acesso

## Como criar *logins* usando T-SQL?

---

- Comando CREATE LOGIN

- Sintaxe básica:

```
create login nome_login  
with password = senha [, default_database = nome_bd]
```

- Caso não seja informado o BD *default* é o master

- Exemplo:

```
use master  
go
```

```
create login aluno  
with password = 's_espb', default_database = pubs  
go
```

- Procedimento armazenado do sistema **sp\_addlogin** (antes do SQL Server 2005).

# Controle de Acesso

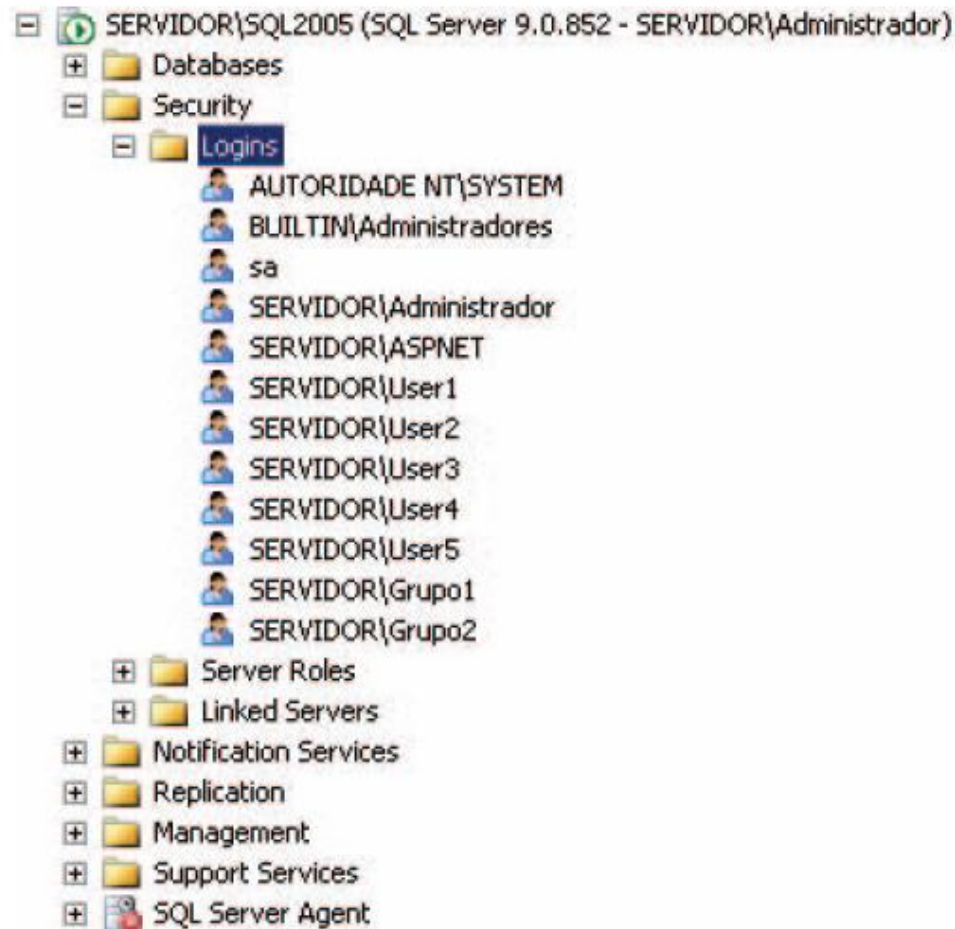
## Como alterar informações e remover *logins*?

---

- Comando ALTER LOGIN
  - Permite alterar informações de um login (senha, bd default etc);
- Comando DROP LOGIN
  - Permite remover um login
- Procedimentos armazenados do sistema **sp\_password**  
**sp\_defaultdb, sp\_droplogin** (antes do SQL Server 2005).

# Controle de Acesso

## *Logins usando o Management Studio*



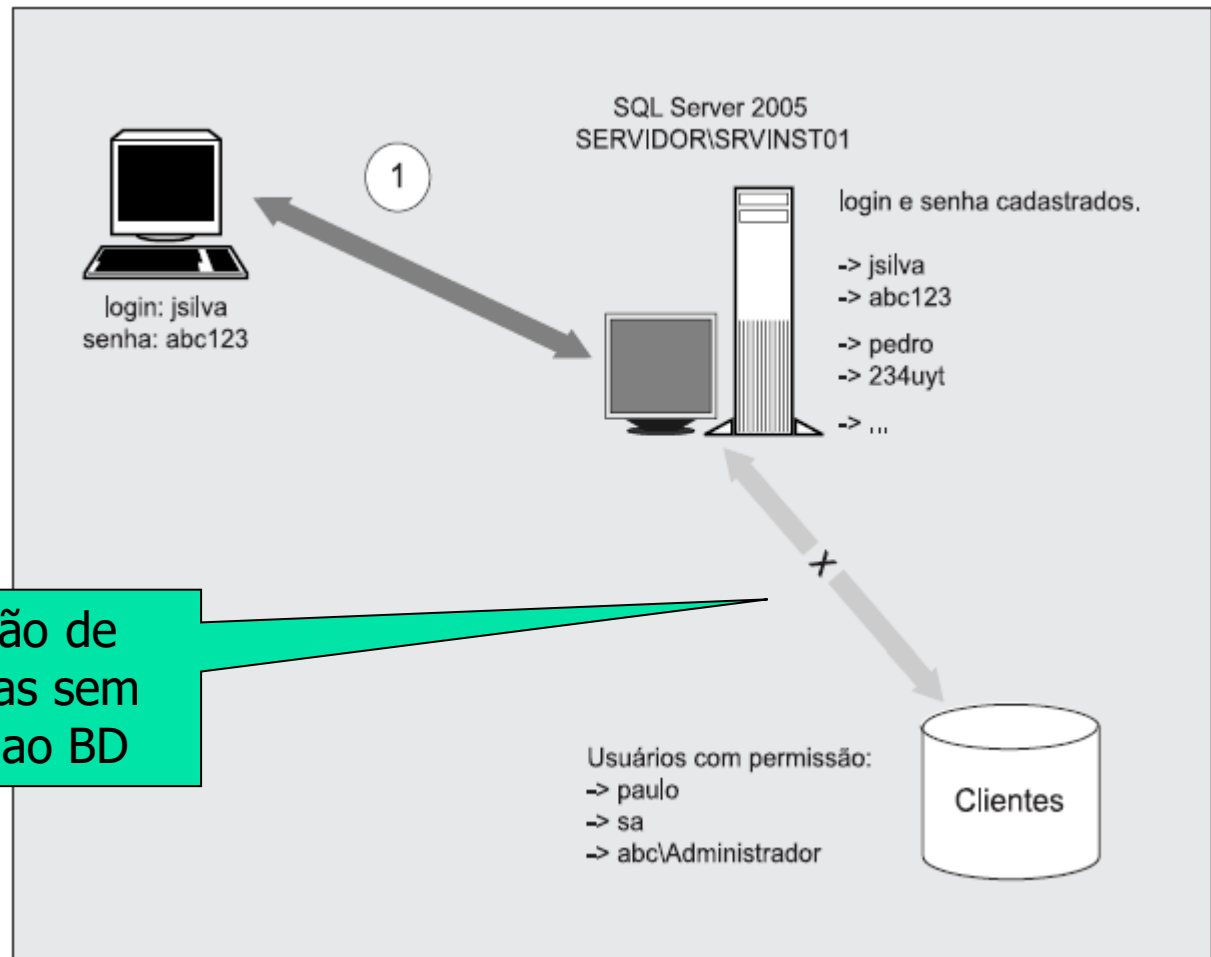


# Controle de Acesso

## Acesso aos BDs de um servidor – *Users*

- Para que um usuário possa utilizar um BD, é necessário que ele possua uma identificação (*user\_id*) no BD em questão.

Usuário com permissão de acesso ao servidor, mas sem permissão de acesso ao BD





# Controle de Acesso

## Como criar *users* usando T-SQL?

---

- Comando CREATE USER

- Sintaxe básica:

```
create user nome_user  
for login nome_login
```

- Exemplo:

```
use bd_espbdb  
go
```

```
create user espdb  
for login aluno  
go
```

- Procedimento armazenado do sistema **sp\_adduser** (antes do SQL Server 2005)



# Controle de Acesso

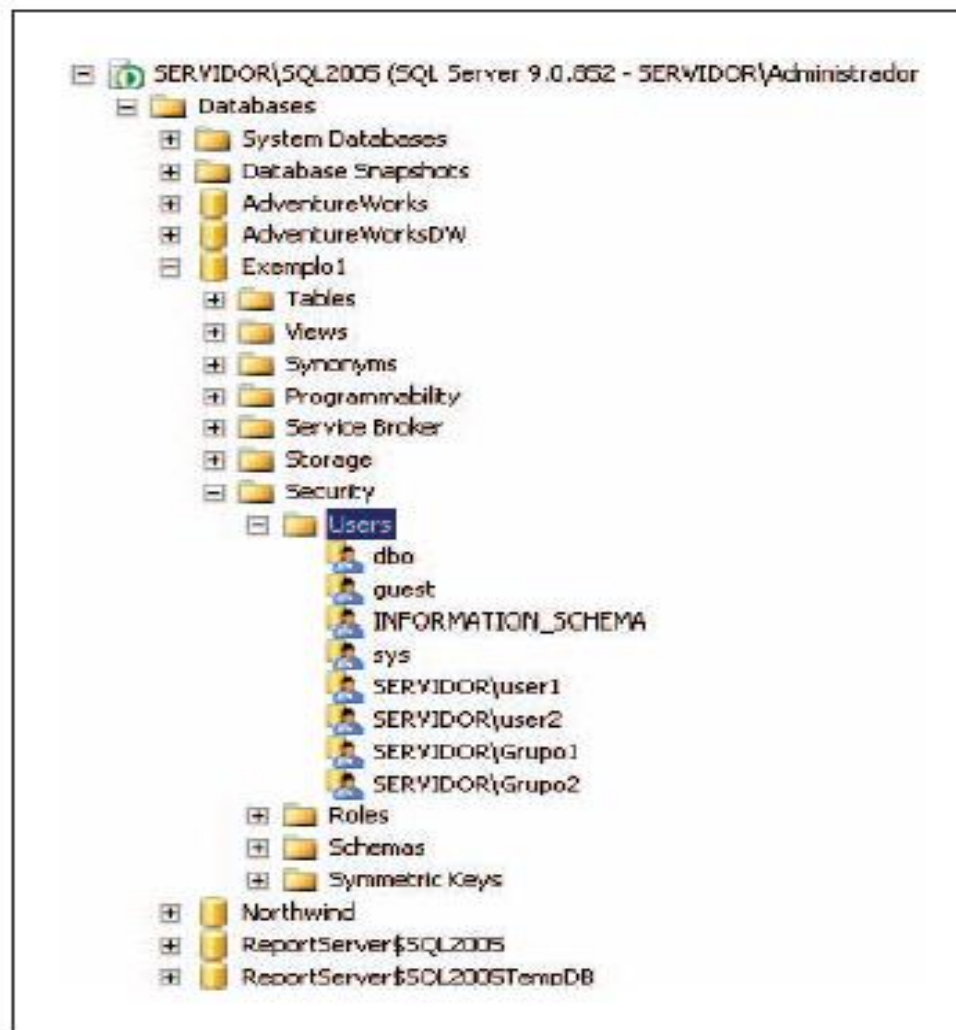
## Como excluir *users* usando T-SQL?

---

- Comando DROP USER
  - Permite remover um user
- Procedimento armazenados do sistema **sp\_dropuser** (antes do SQL Server 2005).

# Controle de Acesso

## *Users usando o Management Studio*





# Controle de Acesso

## *Roles*

---

- ***Roles*** – agrupam usuários (*logins* e *users*) que possuem as mesmas permissões de acesso (conceito de grupos de usuários).
  - Simplificam a atribuição de permissões a usuários: todos os usuários pertencentes a uma *role* herdam as permissões que foram atribuídas para aquela *role*.
- Existem dois tipos de *roles*:
  - *Server Roles*
  - *Database Roles*



# Controle de Acesso

## *Server Roles*

- ***Server Roles*** – têm permissões para realizar tarefas administrativas.
  - Agrupam *logins* do servidor;
  - Existem *server roles* pré-definidas pelo SQL Server. As principais são:

Nome da role	Permissões
sysadmin	Poderes totais sobre todos os objetos da instância.
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 Bancos de Dados.

- A partir do SQL Server 2012 é possível criar novas *server roles* e adicionar permissões do nível de servidor: **CREATE SERVER ROLE**. Em versões anteriores, limita-se a utilização das *server roles* pré-definidas.



# Controle de Acesso

## Como adicionar *logins* a *server roles*?

---

- Comando **ALTER SERVER ROLE**

- Sintaxe básica:

```
alter server role nome_server_role  
    add member nome_login
```

- Exemplo:

```
use master  
go
```

```
alter server role sysadmin  
    add member aluno  
go
```

- Um *login* pode estar associado a uma ou mais *server roles*.
- Procedimento armazenado do sistema **sp\_addsrvrolemember** (antes do SQL Server 2012).



# Controle de Acesso

## Como excluir *logins* de *server roles*?

---

- Comando **ALTER SERVER ROLE**

- Sintaxe básica:

```
alter server role nome_server_role  
drop member nome_login
```

- Exemplo:

```
use master  
go
```

```
alter server role sysadmin  
drop member aluno  
go
```

- Procedimento armazenado do sistema **sp\_dropsrvrolemember** (antes do SQL Server 2012).





# Controle de Acesso

## *Database Roles*

- ***Database Roles*** – têm permissão para realizar tarefas relacionadas aos BDs.
  - Agrupam *users* dos BDs;
  - Existem *database roles* pré-definidas pelo SQL Server. As principais são:

Nome da 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 Banco de Dados.
db_datawriter	Pode adicionar, alterar ou excluir dados em todas as tabelas de usuário do banco de dados.
db_ddladmin	Pode adicionar, modificar ou excluir objetos no Banco de Dados.
db_securityadmin	Pode gerenciar roles e adicionar ou excluir usuários às roles do Banco de Dados. Pode gerenciar as permissões para objetos do Banco de Dados.
db_backupoperator	Pode fazer o backup do Banco de Dados.
db_denydatareader	Não pode consultar dados em nenhuma das tabelas do Banco de Dados, mas pode efetuar alterações na estrutura do Banco de Dados e de seus objetos.
db_denydatawriter	Não pode alterar dados no Banco de Dados.



# Controle de Acesso

## *Database Roles*

---

- *Database role **public**:*
  - Todo *user* de um BD pertence automaticamente a *database role public*,
  - Tem permissões limitadas a leitura de dados.
- É possível criar novas *database roles*: **CREATE ROLE**
  - Sintaxe básica:

```
use nome_bd  
create role nome_database_role
```
- Para excluir *database roles*: **DROP ROLE**
  - Sintaxe básica:

```
use nome_bd  
drop role nome_database_role
```
- Procedimentos armazenados do sistema **sp\_addrole, sp\_droprole** (antes do SQL Server 2005)



# Controle de Acesso

## Como adicionar *users* a *database roles*?

---

- Comando **ALTER ROLE**

- Sintaxe básica:

```
alter role nome_database_role  
    add member nome_user
```

- Exemplo:

```
use bd_espbdb  
go
```

```
alter role db_owner  
    add member espbdb  
go
```

- Um *user* pode estar associado a uma ou mais *database roles*.
- Procedimento armazenado do sistema **sp\_addrolemember** (antes do SQL Server 2012).



# Controle de Acesso

## Como excluir *users* de *database roles*?

---

- Comando **ALTER ROLE**

- Sintaxe básica:

```
alter role nome_database_role  
drop member nome_user
```

- Exemplo:

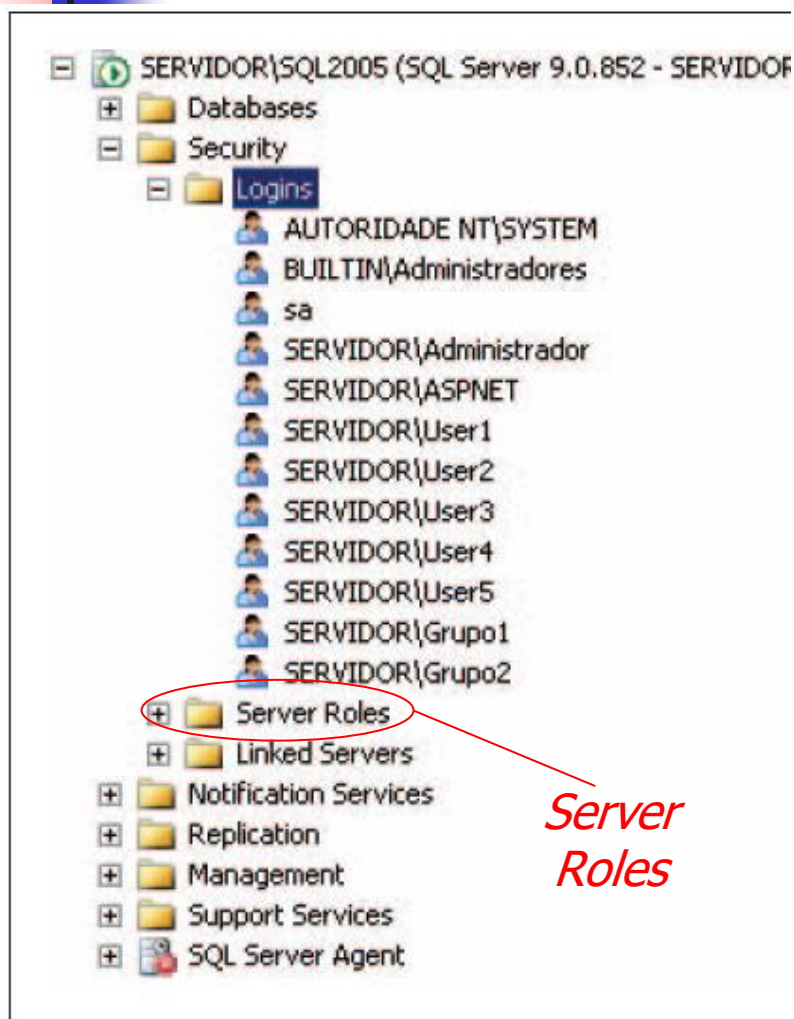
```
use bd_espbdb  
go
```

```
alter role db_owner  
drop member espbdb  
go
```

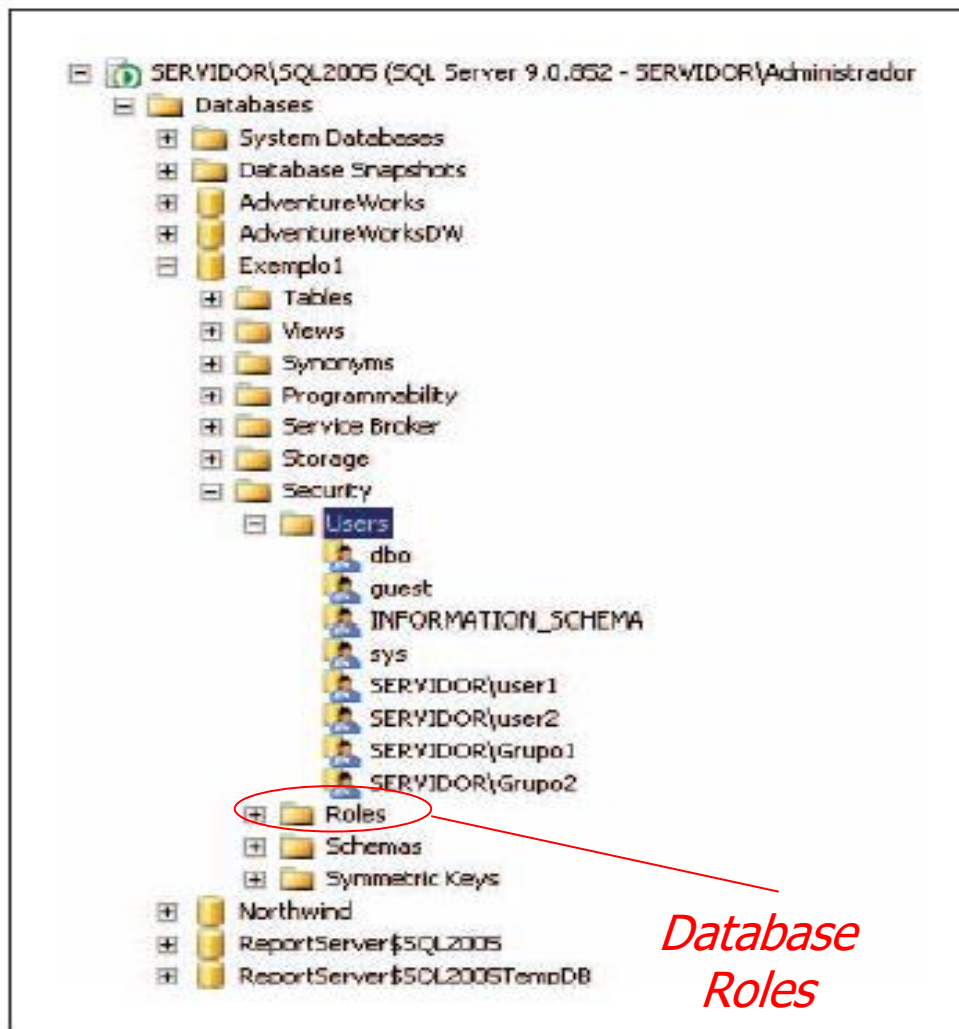
- Procedimento armazenado do sistema **sp\_droprolemember** (antes do SQL Server 2012).

# Controle de Acesso

## *Roles usando o Management Studio*



*Server  
Roles*

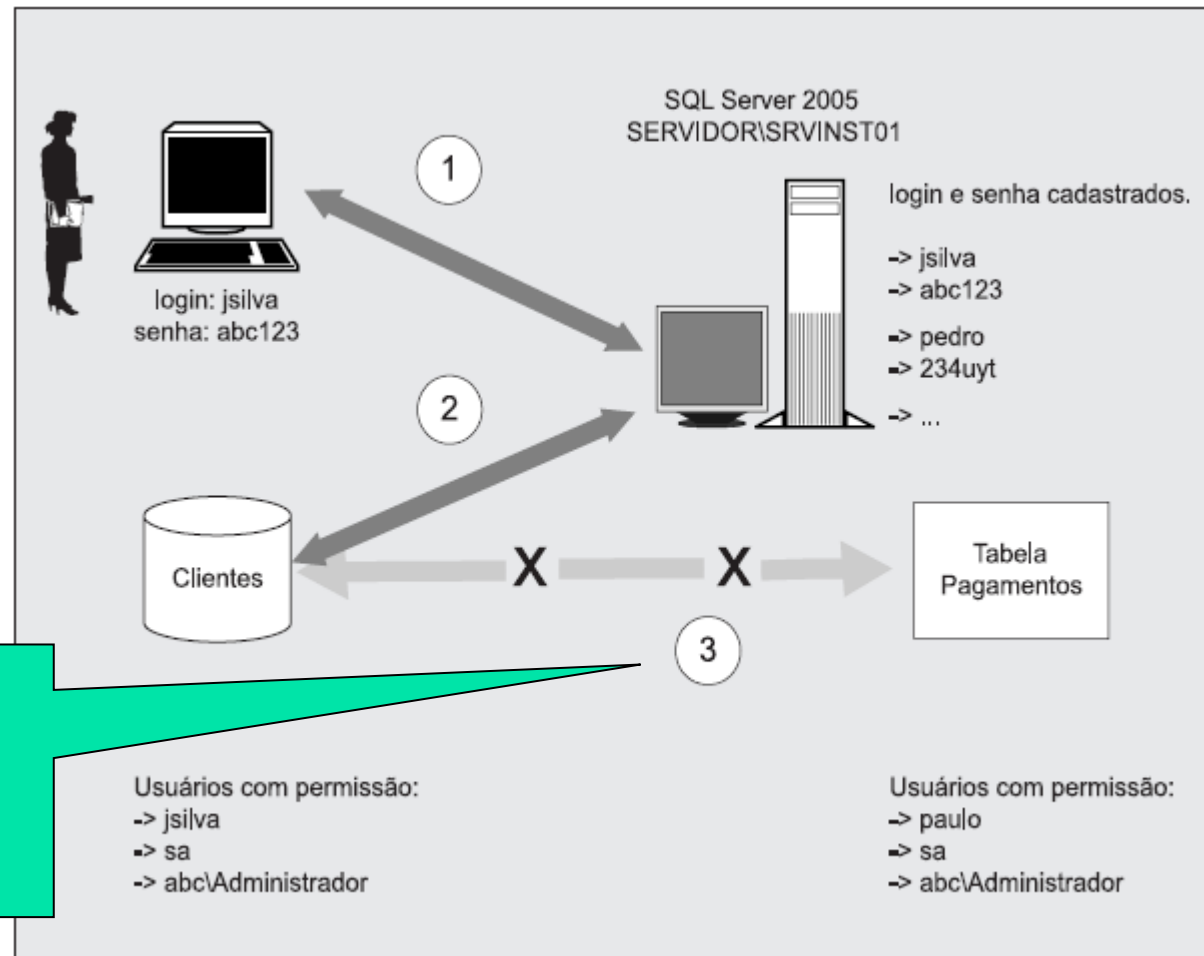


*Database  
Roles*

# Controle de Acesso

## Acesso aos Objetos dos BDs

- Para que um usuário ou *database role* possa utilizar objetos de um BD, é necessário que possua permissões nos objetos em questão.



Usuário com permissão acesso ao servidor e ao BD, mas sem permissão na tabela Pagamentos



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

---

- **Permissões** (ou privilégios) definem quem pode fazer o que em nível de Servidor, de BD e de objetos de BD. Permissões podem ser permitidas ou negadas.
  - **Permissões em nível de servidor** definem o que um *login* pode fazer de acordo com as *server roles* de que for membro.
    - Por exemplo, um *login* membro da *server role* **sysadmin** pode fazer qualquer coisa, tais como criar/excluir/alterar BDs, agendar tarefas, etc;
  - **Permissões em nível de BD** definem o que um *user* ou *database role* pode fazer com um BD.
    - Por exemplo, criar/excluir/alterar qualquer objeto do BD (tabelas, visões, procedimentos armazenados, etc), *backup* do BD, *backup* do *log*, etc.
  - **Permissões em nível de objetos de BDs** definem o que um *user* ou *database role* pode fazer com objetos específicos de um BD.
    - Por exemplo, operações de consulta, inclusão, alteração, exclusão de dados de tabelas, execução de procedimentos armazenados, etc.



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

---

- **Como atribuir/retirar/negar permissões em nível de servidor?**
  - Permitindo ou negando o acesso ao *login*;
  - Colocando/retirando o *login* como membro de uma das *server roles* pré-definidas pelo SQL Server;
  - A partir do SQL Server 2012 é possível também:
    - atribuir/retirar/negar permissões em nível de servidor diretamente ao *login* ou *server role* definida pelo usuário, usando os comandos GRANT/REVOKE/DENY;
    - colocar/retirar o *login* como membro de uma das *server roles* criadas pelo usuário.



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

- **Como atribuir permissões em nível de BD?**

- Colocando o *user* como membro de uma das *database roles* pré-definidas pelo SQL Server ou de *database roles* previamente criadas;
- Atribuindo permissões diretamente ao *user* ou *database role*, usando o comando GRANT:

- Sintaxe básica:

```
grant {all | statement [,...n]}  
to security_account [,...n]
```

todas as permissões

permissões específicas

- Exemplo:

```
use bd_espbdb  
go
```

*user* ou *database role*

```
grant create table, create view  
to espdb
```

- Observação: se for considerado o BD *master*, é possível utilizar o comando GRANT para dar permissões a *users* do BD *master* para CREATE DATABASE.



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

---

- **Principais permissões em nível de BD:**

Permissão	Se aplica a:
CREATE DATABASE	Banco de Dados master.
CREATE DEFAULT	Todos os Bancos de Dados.
CREATE PROCEDURE	Todos os Bancos de Dados.
CREATE RULE	Todos os Bancos de Dados.
CREATE TABLE	Todos os Bancos de Dados.
CREATE VIEW	Todos os Bancos de Dados.
BACKUP DATABASE	Todos os Bancos de Dados.
BACKUP LOG	Todos os Bancos de Dados.



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

---

- **Como retirar permissões em nível de BD?**

- Retirando o *user* de membro de uma das *database roles* pré-definidas pelo SQL Server ou de *database roles* previamente criadas;
- Retirando permissões diretamente ao *user* ou *database role*, usando o comando REVOKE:

- Sintaxe básica:

```
revoke {all | statement [,...n]}  
to security_account [,...n]
```

- Exemplo:

```
use bd_espbdb  
go
```

```
revoke create table, create view  
to espbdb
```



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

### ■ Como atribuir permissões em nível de objetos de BDs?

- Atribuindo permissões diretamente ao *user* ou *database role*, usando o comando GRANT:

- Sintaxe completa:

```
GRANT
{ ALL [ PRIVILEGES ] | permission [ ,...n ] }
{
[ ( column [ ,...n ] ) ] ON { table | view }
| ON { table | view } [ ( column [ ,...n ] ) ]
| ON { stored_procedure | extended_procedure }
| ON { user_defined_function }
}
TO security_account [ ,...n ]
[ WITH GRANT OPTION ]
[ AS { group | role } ]
```

- Exemplo:

```
use bd_espbdb
go
```

```
grant select, update, delete on produto
to espbdb
```



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

---

- **Principais permissões em nível de objetos de BDs:**

Permissão	Se aplica a:
SELECT	Tabelas, views e colunas.
INSERT	Tabelas e views.
DELETE	Tabelas e views.
UPDATE	Tabelas, views e colunas.
EXECUTE	Stored Procedures.



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

### ■ Como retirar permissões em nível de objetos de BDs?

- Retirando permissões diretamente ao *user* ou *database role*, usando o comando REVOKE:

- Sintaxe completa:

```
REVOKE [ GRANT OPTION FOR ]  
{ ALL [ PRIVILEGES ] | permission [ ,...n ] }  
{  
[ ( column [ ,...n ] ) ] ON { table | view }  
| ON { table | view } [ ( column [ ,...n ] ) ]  
| ON { stored_procedure | extended_procedure }  
| ON { user_defined_function }  
}  
{ TO | FROM }  
security_account [ ,...n ]  
[ CASCADE ]  
[ AS { group | role } ]
```

- Exemplo:

```
use bd_espbd  
go
```

```
revoke select, update, delete on produto  
to espbd
```



# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

---

- Uma permissão pode ser **negada**;
- **Negar tem precedência sobre atribuir permissões**;
- **Como negar permissões?**
  - Negando permissões diretamente ao *user* ou *database role*, usando o comando DENY:
  - Sintaxe completa:

```
DENY
{ ALL [ PRIVILEGES ] | permission [ ,...n ] }
{
[ ( column [ ,...n ] ) ] ON { table | view }
| ON { table | view } [ ( column [ ,...n ] ) ]
| ON { stored_procedure | extended_procedure }
| ON { user_defined_function }
}
TO security_account [ ,...n ]
[ CASCADE ]
```

# Controle de Acesso

## Acesso aos Objetos dos BDs – Permissões

- Quais são as permissões efetivas de cada usuário do exemplo seguinte?

<i>Usuários</i>	<i>roles</i>	<i>Membros</i>	<i>Permissões</i>
<ul style="list-style-type: none"><li>• user1</li><li>• user2</li><li>• user3</li><li>• user4</li><li>• user5</li></ul>	role1	<ul style="list-style-type: none"><li>• user1</li><li>• user2</li><li>• user3</li></ul>	<ul style="list-style-type: none"><li>• role1</li><li>• INSERT</li><li>• UPDATE</li></ul>
	role2	<ul style="list-style-type: none"><li>• user3</li><li>• user4</li><li>• user5</li></ul>	<ul style="list-style-type: none"><li>• role2</li><li>• negar INSERT</li><li>• negar DELETE</li><li>• SELECT</li></ul>
	role3	<ul style="list-style-type: none"><li>• user1</li><li>• user3</li><li>• user5</li></ul>	<ul style="list-style-type: none"><li>• role3</li><li>• INSERT</li><li>• DELETE</li><li>• SELECT</li></ul>
	role4	<ul style="list-style-type: none"><li>• user2</li></ul>	<ul style="list-style-type: none"><li>• role4</li><li>• NEGAR UPDATE</li></ul>
	role5	<ul style="list-style-type: none"><li>• ninguém</li></ul>	<ul style="list-style-type: none"><li>• role5</li><li>• negar TUDO</li></ul>
<ol style="list-style-type: none"><li>1. Qual a permissão efetiva do usuário user1 ?</li><li>2. Qual a permissão efetiva do usuário user2 ?</li><li>3. Qual a permissão efetiva do usuário user3 ?</li><li>4. Qual a permissão efetiva do usuário user4 ?</li><li>5. Qual a permissão efetiva do usuário user5 ?</li></ol>			





# Controle de Acesso

## Acesso aos Objetos dos BDs – *Schemas*

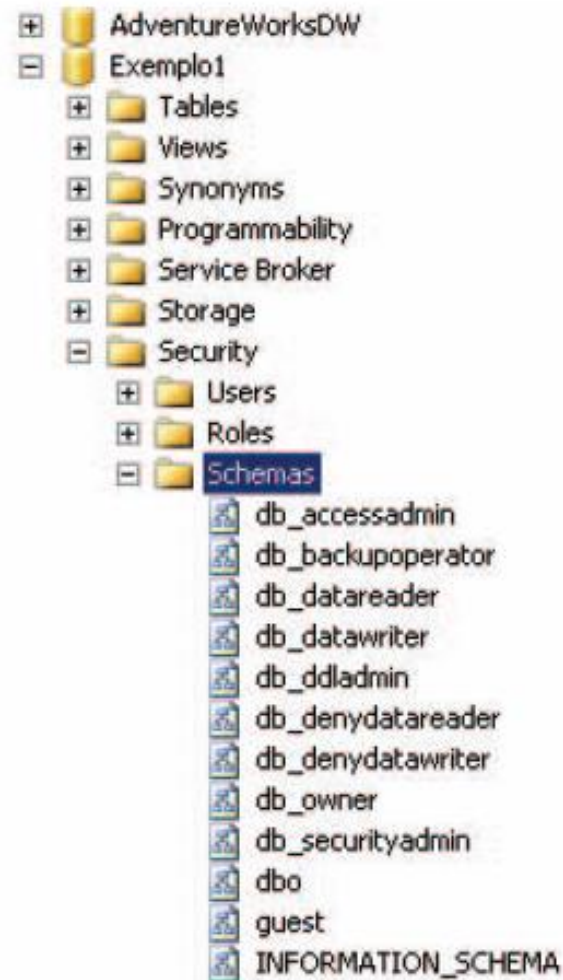
---

- Permissões podem ser atribuídas diretamente a um objeto do BD ou a um ***schema***;
- ***Schema***: agrupa objetos do BD cujo dono é o mesmo *user* ou *role* (*database role*), ou seja, *users* e *roles* passam a ser donos dos *schemas* e não mais de objetos do BD individualmente.
  - *users* e *roles* acessam objetos dos BDs por meio dos *schemas*;
  - vários *users* podem ser donos de um mesmo *schema* por meio da criação de uma *role* para esses *users* e essa *role* é dono do esquema;
  - não pode haver dois ou mais objetos com o mesmo nome em um mesmo *schema*;
  - Nome completo de acesso a objetos de um BD:  
`Nome_Servidor.Nome_BD.Nome_Schema.Nome_Objeto`

# Controle de Acesso

## Acesso aos Objetos dos BDs – *Schemas*

- Pelo *Management Studio* é possível:
  - Criar novos *schemas*;
  - Atribuir objetos a um *schema*;
  - Alterar o dono de um *schema* (deve-se observar que o nome completo de acesso a objetos não muda)





# *Backup/Restore*

---

- Estratégias de *backup/restore* e proteção de dados:
  - Dependem do valor da informação que está sendo protegida:
    - Quais os requisitos de disponibilidade dos dados? – o BD deve estar *on-line* 24 h por dia, 7 dias por semana, ou pode ser colocado *off-line* em alguns horários?
    - Em caso de falhas, qual o tempo aceitável para a volta a normalidade?
  - Possibilidades:
    - Replicação por *mirroring* (“espelhamento”): os dados ficam duplicados em dois servidores e, caso um servidor falhe, o outro assume a função;
    - *Databases snapshots*: cópias do BD em intervalos curtos de tempo;
    - *Backup/Restore*: cópias dos dados para possível restauração posterior.



# *Backup/Restore*

---

- Quanto ao conteúdo do *Backup*, é possível:
  - Fazer *backup* somente do BD:
    - O *backup* do BD é feito periodicamente (pelo menos diariamente);
    - Caso haja algum problema, as alterações ocorridas no BD desde o último backup serão perdidas.
  - Fazer *backup* do BD e do *log* de transações:
    - O *backup* do BD é feito com menos frequência (por exemplo, uma vez por semana);
    - O *backup* do *log* de transações é feito com mais frequência durante o período em que o *backup* do BD não ocorre (por exemplo, 4 vezes ao dia);
    - No caso de algum problema, deve-se restaurar o último *backup* do BD e depois todos os *backups* de *log* de transações em ordem cronológica desde o último *backup* do BD.



# *Backup/Restore*

---

- Tipos de *backup* no SQL Server:
  - ***backup completo – full backup:*** *backup* de todo o BD, incluindo todos os *filegroups* e arquivos de dados que o compõe;
  - ***backup diferencial:*** serão copiados apenas os dados (páginas de dados) alterados desde o último *backup* completo;
  - ***backup do log de transações:*** copia apenas do *log* de transações;
- O processo mais lento é o *backup* completo.



# *Backup/Restore*

## Exemplos de estratégias

---

- *Backup* completo diário do BD:

Dia	Backup/horário(s)
Segunda-feira	Completo às 23h
Terça-feira	Completo às 23h
Quarta-feira	Completo às 23h
Quinta-feira	Completo às 23h
Sexta-feira	Completo às 23h
Sábado	Completo às 23h
Domingo	Completo às 23h



# *Backup/Restore*

## Exemplos de estratégias

---

- *Backup* completo do BD e também *backup* do *log*:

Dia	Backup/horário(s)
Sábado	Completo às 23h
Domingo	Backup do log às 9h, 12h, 15h e 18h
Segunda-feira	Backup do log às 9h, 12h, 15h e 18h
Terça-feira	Backup do log às 9h, 12h, 15h e 18h
Quarta-feira	Backup do log às 9h, 12h, 15h e 18h
Quinta-feira	Backup do log às 9h, 12h, 15h e 18h
Sexta-feira	Backup do log às 9h, 12h, 15h e 18h
Sábado	Completo às 23h



# *Backup/Restore*

## Exemplos de estratégias

- *Backup* completo, diferencial e do *log* de transações:

Dia	Backup/horário(s)
Sábado	Completo às 23h
Domingo	Backup do log às 9h, 12h, 15h e 18h
Segunda-feira	Backup do log às 9h, 12h, 15h e 18h
Terça-feira	Backup do log às 9h, 12h, 15h e 18h
Quarta-feira	Backup do log às 9h, 12h, 15h e 18h Backup diferencial às 23h
Quinta-feira	Backup do log às 9h, 12h, 15h e 18h
Sexta-feira	Backup do log às 9h, 12h, 15h e 18h
Sábado	Diferencial às 23h
Domingo	Backup do log às 9h, 12h, 15h e 18h
Segunda-feira	Backup do log às 9h, 12h, 15h e 18h
Terça-feira	Backup do log às 9h, 12h, 15h e 18h
Quarta-feira	Backup do log às 9h, 12h, 15h e 18h Backup diferencial às 23h
Quinta-feira	Backup do log às 9h, 12h, 15h e 18h
Sexta-feira	Backup do log às 9h, 12h, 15h e 18h
Sábado	Diferencial às 23h





# *Backup/Restore*

---

- Quando fazer *backup* do BD *master*?
  - Regularmente;
  - Imediatamente após:
    - Criação/alteração/exclusão de BDs;
    - Criação/exclusão de *logins*, *users*, *roles* e atribuição/exclusão/negação de permissões.



# *Backup/Restore*

---

- Novidades sobre *backups* a partir do SQL Server 2005:
  - ***Partial Restore***: durante o processo de *restore* o BD fica disponível (*on-line*) para uso logo após a restauração do *filegroup primary*.
    - Nas versões anteriores do SQL Server, durante todo o processo de *restore* o BD fica indisponível (*off-line*);
  - ***Concurrent Databases and Log Backups***: realização simultânea do *backup* completo do BD e do *backup* do *log* de transações.
    - Nas versões anteriores do SQL Server, o *backup* completo do BD tem que ter sido finalizado para poder iniciar o *backp* do *log* de transações.



# *Backup/Restore*

## Como fazer um *backup*?

---

- Para fazer o *backup* do BD:
  - Pelo *Management Studio*: Nome do BD -> Tasks -> Back up
    - Opção *backup type*: *full*
  - Em T-SQL usa-se o comando **BACKUP DATABASE**
- Para fazer o *backup* do *log* de transações:
  - Pelo *Management Studio*: Nome do BD -> Tasks -> Back up
    - Opção *backup type*: *transaction log*
  - Em T-SQL usa-se o comando **BACKUP LOG**



# *Backup/Restore*

## Como fazer um *restore*?

---

- *Restore* do BD:

- Pelo *Management Studio*: Nome do BD -> Tasks -> Restore -> Database
- Em T-SQL usa-se o comando **RESTORE DATABASE**

- *Restore* do *Log* de transações:

- Pelo *Management Studio*: Nome do BD -> Tasks -> Restore -> Database
- Em T-SQL usa-se o comando **RESTORE LOG**



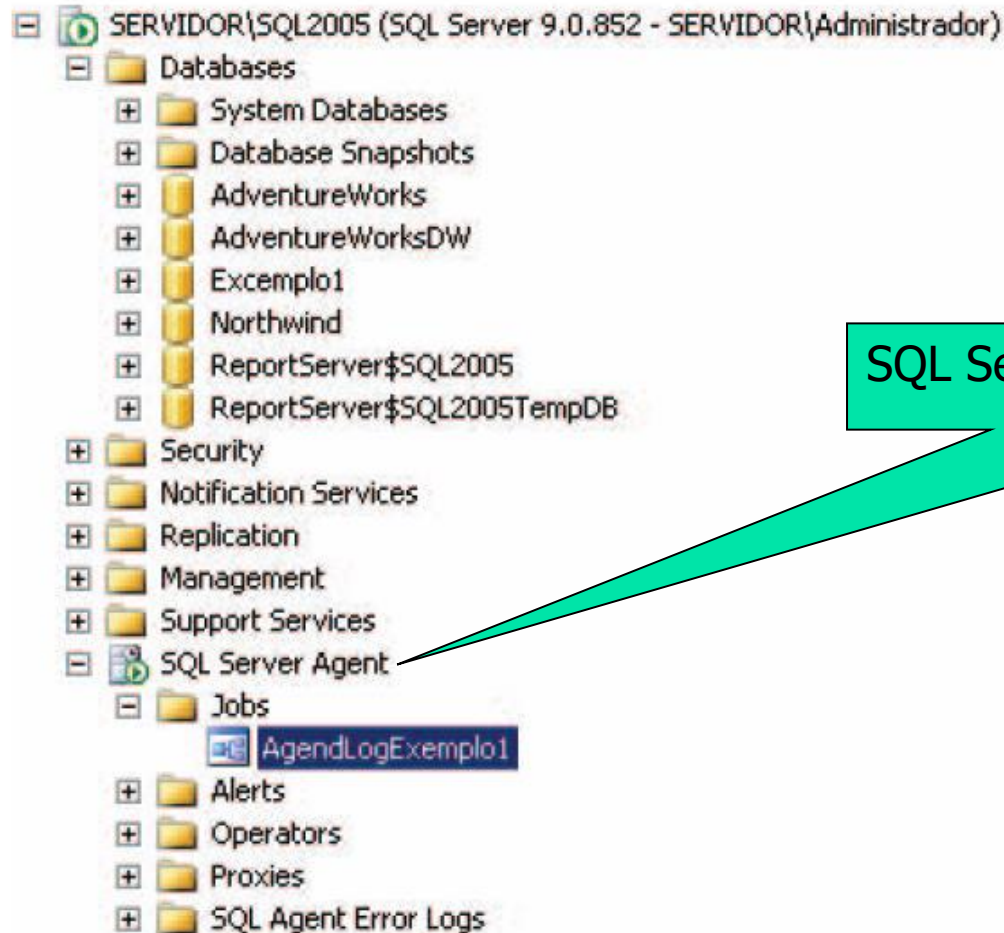
# Agendamento de Tarefas – *Jobs*

---

- *Job* – tarefa administrativa, composta por um ou mais passos, a qual é agendada para executar automaticamente, em datas e horários determinados em sua configuração.
  - Tarefas administrativas: tarefas rotineiras que o DBA deve executar como *backups*, manutenção de índices e outras tarefas que necessitem ser executadas periodicamente;
  - Um *Job* também pode ser criado e não agendado, a fim de ser executado manualmente;
- O serviço *SQL Server Agent* deve estar rodando;
- Usa o BD do sistema *msdb* que armazena os *Jobs*.

# Agendamento de Tarefas – *Jobs*

## Como criar *Jobs*?



SQL Server Agent -> Jobs -> New Job

# Agendamento de Tarefas – *Jobs*

## Como criar *Jobs*?

- Janela geral para criação do *Job*

The screenshot shows the 'New Job' dialog box in SQL Server Enterprise Manager. The window has a title bar 'New Job' and standard Windows window controls. On the left, there is a 'Select a page' sidebar with a tree view containing 'General', 'Steps', 'Schedules', 'Alerts', 'Notifications', and 'Targets'. The 'General' page is selected. Below the sidebar, there are sections for 'Connection' and 'Progress'. The 'Connection' section shows 'Server: SERVIDOR\SQL2005' and 'Connection: SERVIDOR\Administrador', with a link to 'View connection properties'. The 'Progress' section shows a circular progress indicator and the text 'Ready'. The main area of the dialog contains fields for 'Name', 'Owner', 'Category', and 'Description'. The 'Name' field is 'BackupExemplo1AdventureWorks', 'Owner' is 'sa', and 'Category' is '[Uncategorized (Local)]'. The 'Description' field contains the text 'Faz o backup de Exemplo1 e AdventureWorks, diariamente, às 23:00 h.'. There is an 'Enabled' checkbox which is checked. At the top of the main area, there are buttons for 'Refresh', 'Schedule', 'Script', and 'Help'. At the bottom right, there are 'OK' and 'Cancel' buttons.

**New Job**

Select a page

- General
- Steps
- Schedules
- Alerts
- Notifications
- Targets

Connection

Server: SERVIDOR\SQL2005

Connection: SERVIDOR\Administrador

[View connection properties](#)

Progress

Ready

Name: BackupExemplo1AdventureWorks

Owner: sa

Category: [Uncategorized (Local)]

Description: Faz o backup de Exemplo1 e AdventureWorks, diariamente, às 23:00 h.

☒ Enabled

Refresh Schedule Script Help

OK Cancel

# Agendamento de Tarefas – *Jobs*

## Como criar *Jobs*?

- Janela para criação de passos do *Job* (opção Steps da janela anterior)

The screenshot shows the 'New Job Step' dialog box. On the left, there is a 'Select a page' section with 'General' and 'Advanced' tabs. Below this is a 'Connection' section showing 'Server: SERVIDOR\SQL2005' and 'Connection: SERVIDOR\Administrador', with a link to 'View connection properties'. At the bottom left is a 'Progress' section with a circular progress indicator and the word 'Ready'. The main area on the right contains the following fields and controls:

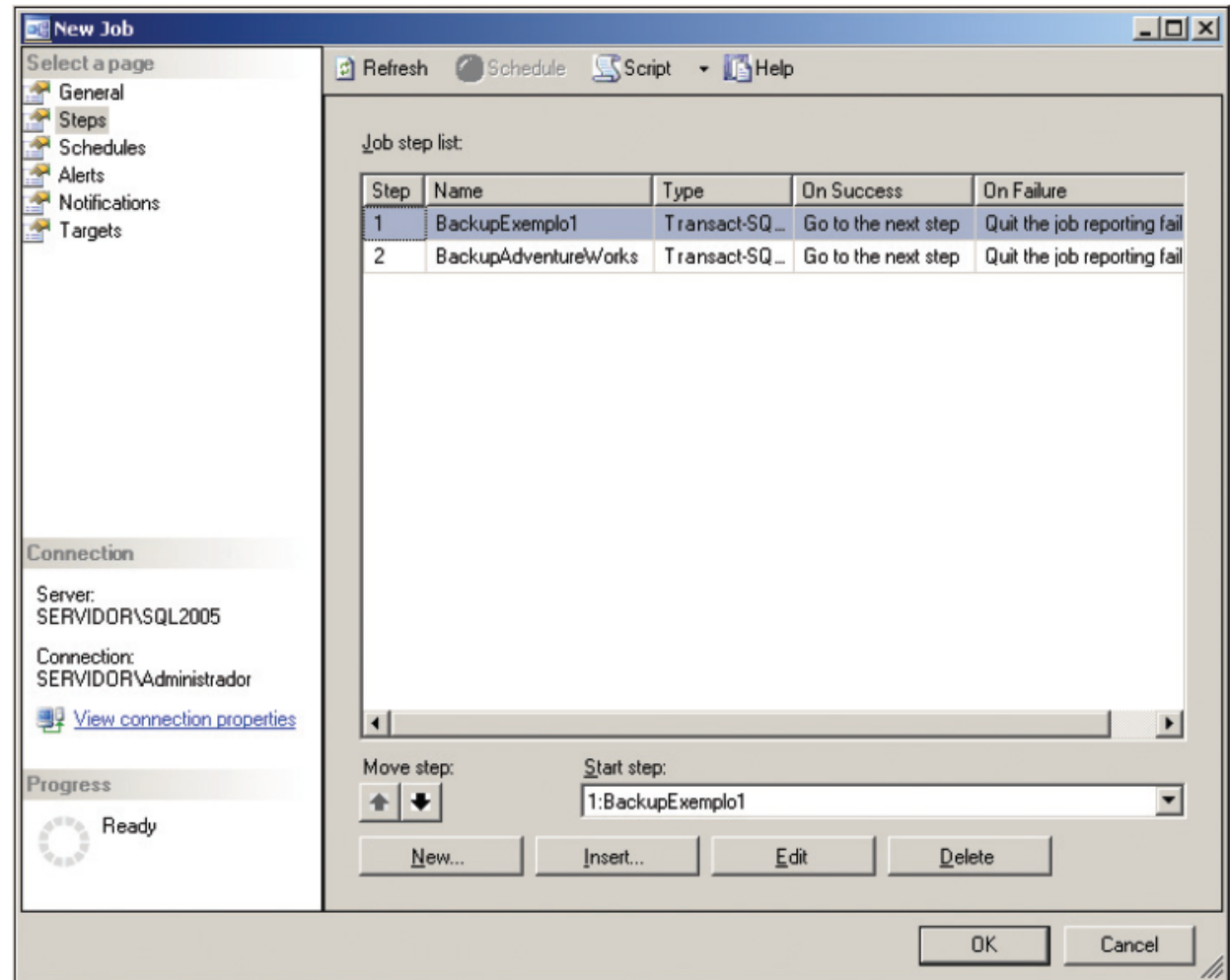
- Step name:** A text input field.
- Type:** A dropdown menu currently set to 'Transact-SQL script (T-SQL)'.
- Run as:** A dropdown menu.
- Database:** A dropdown menu currently set to 'master'.
- Command:** A large text area for entering the command.
- Buttons below the Command area: 'Open...', 'Select All', 'Copy', 'Paste', and 'Parse'.
- Buttons at the bottom right: 'OK' and 'Cancel'.



# Agendamento de Tarefas – *Jobs*

## Como criar *Jobs*?

- Dois passos já adicionados ao *Job*.



# Agendamento de Tarefas – *Jobs*

## Como criar *Jobs*?

- Configurando o agendamento do *Job* – Guia *Schedules* da janela de criação do *Job*.

**New Job Schedule**

Name:

Schedule Type:  ☒ Enabled

One time occurrence

On date:  At time:

Frequency

Occur:

☒ Day  of every  month(s)

☐ The   of every  month(s)

Daily frequency

☒ Occurs once at

☐ Occurs every  hour(s) Starting at:  Ending at:

Duration

Start date:  ☐ End date:

☒ No end date:

Summary

Description:



# Bibliografia

---

## Login no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/security/authentication-access/create-a-login?view=sql-server-ver15>

## User no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver15>

## Roles no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/security/authentication-access/server-level-roles?view=sql-server-ver15>

<https://docs.microsoft.com/pt-br/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-ver15>

## Permissões no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/security/authentication-access/determining-effective-database-engine-permissions?view=sql-server-ver15>



# Bibliografia

---

## Schemas no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/security/authentication-access/create-a-database-schema?view=sql-server-ver15>

## Backup e Restore no SQL Server

<https://docs.microsoft.com/pt-br/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases?view=sql-server-ver15>

<https://docs.microsoft.com/pt-br/sql/relational-databases/backup-restore/back-up-a-transaction-log-sql-server?view=sql-server-ver15#using-transact-sql>

## Jobs no SQL Server

<https://docs.microsoft.com/pt-br/sql/ssms/agent/schedule-a-job?view=sql-server-ver15>