

Mão na massa

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

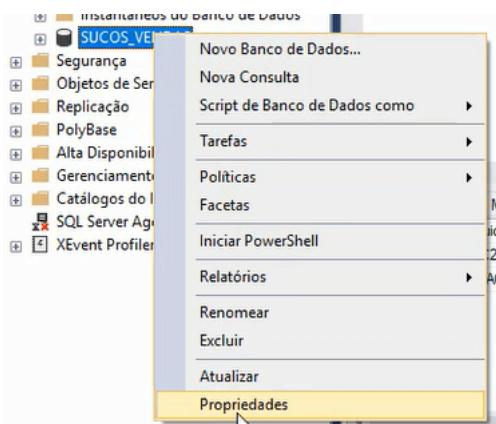
1) Vá ao **SQL Server Management Studio** e crie uma nova consulta.

2) Digite os comandos abaixo:

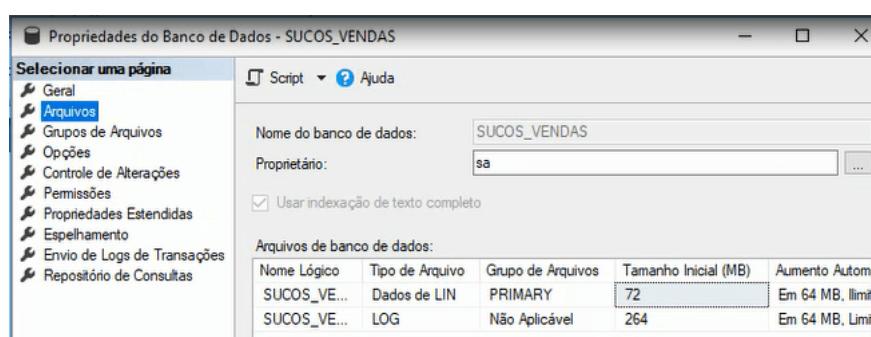
```
USE SUCOS_VENDAS
SELECT * FROM SYS.database_files
```

O comando acima mostra os arquivos físicos que compõem o banco de dados **SUCOS_VENDAS**. Ele sempre irá apresentar os arquivos do banco de dados que está sendo usado no momento.

3) Clique com botão da direita do mouse sobre o nome da base de dados e escolha **Propriedades**:



Depois, escolha a seção **Arquivos**, você verá as mesmas informações exibidas pelo comando T-SQL do item anterior:



4) Se você digitar o comando abaixo:

```
CREATE DATABASE newDB
```

Uma nova base será criada usando os padrões contidas da base **MODEL**.

5) Você pode verificar como e onde foi criada a base **newDB**. Para isso, digite:

```
USE newDB
SELECT * FROM SYS.database_files
```

A base newDB terá um arquivo primário chamado NEWDB.MDF e um de log chamado NEWDB_LOG.LDF .

6) Você pode criar bases utilizando parâmetros fora do padrão da base MODEL . Para isso, você tem que incluir os parâmetros de propriedades no próprio comando. Para isso, certifique-se que na sua máquina você tem o diretório C:\TEMP\SQL (se não tiver, crie-o) e, com o comando abaixo, crie uma base cujos arquivos de dados e log estarão neste novo diretório:

```
CREATE DATABASE ARCHIVE
ON PRIMARY (NAME = ARCH1, FILENAME = 'C:\Temp\SQL\archive1.mdf', SIZE = 100MB,
MAXSIZE = 200MB, FILEGROWTH = 20MB),
(NAME = ARCH2, FILENAME = 'C:\Temp\SQL\archive2.ndf', SIZE = 100MB,
MAXSIZE = 200MB, FILEGROWTH = 20MB)
LOG ON (NAME = ARCH1LOG, FILENAME = 'C:\Temp\SQL\archive2.ldf', SIZE = 100MB,
MAXSIZE = 200MB, FILEGROWTH = 20MB)
```

7) Você pode, depois, verificar se os arquivos referentes à base acima foram criados no sistema operacional:

Este Computador > Disco Local (C:) > Temp > SQL				
	Nome	Data de modific...	Tipo	Tamanho
es	archive1.mdf	09/07/2018 14:43	SQL Server Database...	102.400 KB
	archive2.ldf	09/07/2018 14:43	SQL Server Database...	102.400 KB
	archive2.ndf	09/07/2018 14:43	SQL Server Database...	102.400 KB

8) Você pode reduzir o tamanho da base de dados, usando o conjunto de comandos abaixo:

```
USE SUCOS_VENDAS
ALTER DATABASE SUCOS_VENDAS SET RECOVERY SIMPLE
DBCC SHRINKDATABASE ('SUCOS_VENDAS', NOTRUNCATE)
DBCC SHRINKDATABASE ('SUCOS_VENDAS', TRUNCATEONLY)
ALTER DATABASE SUCOS_VENDAS SET RECOVERY FULL
```

Antes de executá-los, verifique, no diretório do SQL Server, onde os arquivos físicos da base SUCOS_VENDAS foram recuperados.

9) O diretório onde os arquivos estão pode ser verificado pelos seguintes comandos:

```
USE SUCOS_VENDAS
SELECT * FROM SYS.database_files
```

E o padrão seria: c:\Program Files Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA .

10) Verifique o tamanho dos arquivos:

SUCOS_VENDAS.mdf	09/07/2018 14:33	SQL Server Database...	73.728 KB
SUCOS_VENDAS_log.ldf	09/07/2018 14:33	SQL Server Database...	270.336 KB

Agora, execute os comandos iniciais desta seção:

```
USE SUCOS_VENDAS
ALTER DATABASE SUCOS_VENDAS SET RECOVERY SIMPLE
DBCC SHRINKDATABASE ('SUCOS_VENDAS', NOTRUNCATE)
DBCC SHRINKDATABASE ('SUCOS_VENDAS', TRUNCATEONLY)
ALTER DATABASE SUCOS_VENDAS SET RECOVERY FULL
```

Verificando os arquivos novamente, note que houve uma redução no tamanho:

SUCOS_VENDAS.mdf	09/07/2018 14:47	SQL Server Database...	18.496 KB
SUCOS_VENDAS_log.ldf	09/07/2018 14:47	SQL Server Database...	8.192 KB

11) Crie uma nova consulta no **SQL Server Management Studio** e digite os comandos abaixo:

```
USE SUCOS_VENDAS
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'c:\temp\sql\SUCOS_VENDAS.BAK'
```

Neste comando, você irá gerar um *full backup* chamado `SUCOS_VENDAS.BAK` no diretório `C:\TEMP\SQL` (se você não possuir este diretório na sua máquina, crie-o):

archive1.mdf	09/07/2018 14:43	SQL Server Database...	102.400 KB
archive2.ldf	09/07/2018 14:43	SQL Server Database...	102.400 KB
archive2.ndf	09/07/2018 14:43	SQL Server Database...	102.400 KB
SUCOS_VENDAS.BAK	09/07/2018 13:13	Arquivo BAK	19.543 KB

12) Repetindo o comando, mas agora com o parâmetro `INIT`:

```
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'c:\temp\sql\SUCOS_VENDAS.BAK' WITH INIT
```

Significa que o arquivo possuirá dois segmentos (sejam backups parciais ou de logs) no mesmo arquivo `.BAK` e que aquele é o inicial. Se houvessem, antes, neste arquivo `.BAK`, outros segmentos de backup, eles seriam apagados.

13) Aqui, crie o backup, mas comprimido:

```
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'c:\temp\sql\SUCOS_VENDAS2.BAK' WITH COMPRESSION
```

14) Neste ponto, execute um comando qualquer na sua base (por exemplo, abaixo são geradas novas notas fiscais usando uma *Stored Procedure*) para forçar a geração de log de transação. Depois, gere, no mesmo backup inicial, mais um segmento:

```
EXEC [dbo].[CriaNotaFiscal] '20180401'
BACKUP LOG SUCOS_VENDAS TO DISK = 'c:\temp\sql\SUCOS_VENDAS_LOG.BAK'
```

15) Podemos gerar um backup comprimido de log de transação:

```
BACKUP LOG SUCOS_VENDAS TO DISK = 'c:\temp\sql\SUCOS_VENDAS2_LOG.BAK' WITH COMPRESSION
```

16) Você pode gerar um backup de log de transação em um outro arquivo:

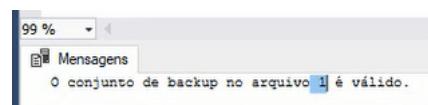
```
EXEC [dbo].[CriaNotaFiscal] '20180402'
BACKUP LOG SUCOS_VENDAS TO DISK = 'c:\temp\sql\SUCOS_VENDAS_P2_LOG.BAK'
```

17) Finalmente, você visualiza os arquivos gerados no diretório:

archive1.mdf	09/07/2018 14:43	SQL Server Database...	102.400 KB
archive2.ldf	09/07/2018 14:43	SQL Server Database...	102.400 KB
archive2.ndf	09/07/2018 14:43	SQL Server Database...	102.400 KB
SUCOS_VENDAS.BAK	09/07/2018 13:14	Arquivo BAK	19.543 KB
SUCOS_VENDAS_LOG.BAK	09/07/2018 13:16	Arquivo BAK	85 KB
SUCOS_VENDAS_P2_LOG.BAK	09/07/2018 13:18	Arquivo BAK	85 KB
SUCOS_VENDAS2.BAK	09/07/2018 13:14	Arquivo BAK	4.194 KB
SUCOS_VENDAS2_LOG.BAK	09/07/2018 13:17	Arquivo BAK	7 KB

18) Crie uma nova consulta no **SQL Server Management Studio** e verifique a integridade do backup, executando:

```
RESTORE VERIFYONLY FROM DISK = 'C:\TEMP\SUCOS_VENDAS.BAK'
```



19) Você também pode verificar os segmentos de backups salvos:

```
RESTORE HEADERONLY FROM DISK = 'C:\TEMP\SUCOS_VENDAS.BAK'
```

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	Owner
1	NULL	1	NULL	0	1	2	sa	sa

20) Salve um backup da sua base de dados SUCOS_VENDAS :

```
ALTER DATABASE SUCOS_VENDAS SET RECOVERY FULL
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\TEMP\SUCOS_VENDAS3.BAK' WITH INIT
```

21) Agora, verifique se existem notas fiscais para o dia 04/04/2018:

```
SELECT * FROM [NOTAS FISCAIS] WHERE DATA = '20180404'
```

A consulta acima deve retornar vazia:

CPF	MATRICULA	DATA	NUMERO	IMPOSTO

22) Então, gere algumas notas fiscais para esta data. Depois, consulte para ver se agora retorna alguma coisa:

```
EXEC [dbo].[CriaNotaFiscal] '20180404'
SELECT * FROM [NOTAS FISCAIS] WHERE DATA = '20180404'
```

	CPF	MATRICULA	DATA	NUMERO	IMPOSTO
1	2600586709	00237	2018-04-05	88023	0,18
2	8719655770	00236	2018-04-05	88024	0,18
3	9283760794	00236	2018-04-05	88025	0,18
4	19290992743	00235	2018-04-05	88026	0,18

23) Agora, recupere o backup que você salvou antes de inserir notas fiscais para o dia 04/04/2018:

```
USE MASTER
ALTER DATABASE SUCOS_VENDAS SET SINGLE_USER WITH ROLLBACK IMMEDIATE
DROP DATABASE SUCOS_VENDAS
RESTORE DATABASE SUCOS_VENDAS FROM DISK = 'C:\TEMP\SUCOS_VENDAS3.BAK' WITH RECOVERY
```

24) Se você executar a consulta, mais uma vez você não verá mais notas fiscais para o dia 04/04/2018:

```
SELECT * FROM [NOTAS FISCAIS] WHERE DATA = '20180404'
```

CPF	MATRICULA	DATA	NUMERO	IMPOSTO

25) Uma política de backup em uma empresa consiste em salvarmos *full backups*, diferenciais e de log em uma sequência no mesmo arquivo de backup, como mostrado abaixo:

```
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\TEMP\SUCOS_VENDAS4.BAK' WITH INIT -- INICIAL
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\TEMP\SUCOS_VENDAS4.BAK' -- BACKUP 2
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\TEMP\SUCOS_VENDAS4.BAK' -- BACKUP 3
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\TEMP\SUCOS_VENDAS4.BAK' -- BACKUP 4
```

26) O comando abaixo permite que você possa conferir os segmentos de backup salvos:

```
RESTORE HEADERONLY FROM DISK = 'C:\TEMP\SUCOS_VENDAS4.BAK'
```

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	Database
1	NULL	1	NULL	0	1	2	sa	DESKTOP-N51LB16	SUCOS_
2	NULL	1	NULL	0	2	2	sa	DESKTOP-N51LB16	SUCOS_
3	NULL	1	NULL	0	3	2	sa	DESKTOP-N51LB16	SUCOS_
4	NULL	1	NULL	0	4	2	sa	DESKTOP-N51LB16	SUCOS_

27) Como todos se tratam de *full backups*, se você quiser retornar o backup do momento em que salvou o quarto arquivo de backup, basta executar:

```
USE MASTER
ALTER DATABASE SUCOS_VENDAS SET SINGLE_USER WITH ROLLBACK IMMEDIATE
DROP DATABASE SUCOS_VENDAS
RESTORE DATABASE SUCOS_VENDAS FROM DISK = 'C:\TEMP\SUCOS_VENDAS4.BAK' WITH FILE = 4, RECOVERY
```

28) Crie uma nova consulta no **SQL Server Management Studio**. Gere um processo simulando uma política de backup.

Primeiro, crie uma tabela que irá gerar marcadores para que você possa identificar diferenças no conteúdo da base de dados entre cada geração de um segmento de backup:

```
USE SUCOS_VENDAS
CREATE TABLE TST_BACKUP (POS INT)
INSERT INTO TST_BACKUP (POS) VALUES (1)
SELECT * FROM TST_BACKUP
```

Durante cada segmento de backup, grave na tabela `TST_BACKUP` um valor que identifica o segmento de backup que será salvo.

29) Inicie a política de backup. Às 1:00 AM, crie um *full backup* com a propriedade `INIT`, para forçar a situação de que se trata de um backup inicial:

```
-- 1:00 AM -- BACKUP FULL INICIAL
INSERT INTO TST_BACKUP (POS) VALUES (1)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH INIT
```

Não esquecendo de incluir no marcador o número 1.

30) Repita, a cada duas horas, a partir das 4:00 AM, backups de log de transação, até as 8:00 AM:

```
-- 4:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (2)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT
```

```
-- 6:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (3)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT
```

```
-- 8:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (4)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT
```

31) Às 9:00 AM, você terá um backup diferencial. Ele irá fazer uma cópia desde o primeiro *full* até agora:

```
-- 9:00 AM - BACKUP DIFERENCIAL
INSERT INTO TST_BACKUP (POS) VALUES (5)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH DIFFERENTIAL
```

32) Novos backups de log, a cada duas horas:

```
-- 10:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (6)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
```

```

WITH NOINIT

-- 12:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (7)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 14:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (8)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

```

33) Mais um backup diferencial:

```

-- 14:00 PM - BACKUP DIFERENCIAL
INSERT INTO TST_BACKUP (POS) VALUES (9)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH DIFFERENTIAL

```

34) De novo, backups de logs:

```

-- 17:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (11)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 19:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (12)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 21:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (13)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

```

35) Simule a passagem do dia. Para isso, execute os comandos acima parcialmente.

Primeiro, selecione os comandos do backup inicial, como se você estivesse às 1:00 AM. Execute os comandos:

```

-- 1:00 AM -- BACKUP FULL INICIAL
INSERT INTO TST_BACKUP (POS) VALUES (1)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH INIT

```

Depois, vá executando cada grupo de comandos até o último, que é o backup de log de 21:00 PM.

36) Se você executar a consulta abaixo:

```
SELECT * FROM TST_BACKUP
```

Você verá uma sequência de 1 a 13 mostrando que todos os backups foram executados durante o dia:

POS	
7	7
8	8
9	9
10	10
11	11
12	12
13	13

37) Já executando:

```
RESTORE HEADERONLY FROM DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
```

Você verá os segmentos de backup contidos em `SUCOS_VENDAS_COMPLETO.BAK`. Você terá um *full backup*, dois diferenciais e diversos de log:

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseNa
1	NULL	1	NULL	0	1	2	sa	DESKTOP-N51LB16	SUCOS_VE
2	NULL	2	NULL	0	2	2	sa	DESKTOP-N51LB16	SUCOS_VE
3	NULL	2	NULL	0	3	2	sa	DESKTOP-N51LB16	SUCOS_VE
4	NULL	2	NULL	0	4	2	sa	DESKTOP-N51LB16	SUCOS_VE
5	NULL	5	NULL	0	5	2	sa	DESKTOP-N51LB16	SUCOS_VE
6	NULL	2	NULL	0	6	2	sa	DESKTOP-N51LB16	SUCOS_VE
7	NULL	2	NULL	0	7	2	sa	DESKTOP-N51LB16	SUCOS_VE
8	NULL	2	NULL	0	8	2	sa	DESKTOP-N51LB16	SUCOS_VE
9	NULL	5	NULL	0	9	2	sa	DESKTOP-N51LB16	SUCOS_VE
10	NULL	2	NULL	0	10	2	sa	DESKTOP-N51LB16	SUCOS_VE
11	NULL	2	NULL	0	11	2	sa	DESKTOP-N51LB16	SUCOS_VE

38) O seu objetivo agora é recuperar o backup de 17:30 PM. Para isso, você precisa identificar a sequência de backups para saber a ordem que deve ser recuperada.

Primeiro, deixe a base indisponível para acesso e depois apague o banco:

```
USE MASTER
ALTER DATABASE SUCOS_VENDAS SET SINGLE_USER WITH ROLLBACK IMMEDIATE
DROP DATABASE SUCOS_VENDAS
```

39) O *full backup* inicial do dia será o ponto de partida:

```
-- 1:00 AM -- BACKUP FULL INICIAL (1)
RESTORE DATABASE SUCOS_VENDAS FROM DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH FILE = 1, NORECOVERY
```

40) Depois, inicie a recuperação usando o primeiro diferencial antes de 17:30. Será gerado às 14:00 PM:

```
-- 14:00 PM - BACKUP DIFERENCIAL (9)
RESTORE DATABASE SUCOS_VENDAS FROM DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH FILE = 9, NORECOVERY
```

41) Então, na sequência, vá recuperando os arquivos de log das 15:00 PM e 17:00 PM:

```
-- 15:00 PM - BACKUP LOG (10)
RESTORE LOG SUCOS_VENDAS FROM DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH FILE = 10, NORECOVERY

-- 17:00 PM - BACKUP LOG (11)
RESTORE LOG SUCOS_VENDAS FROM DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH FILE = 11, RECOVERY
```

42) Verifique novamente o arquivo de marcação:

```
USE SUCOS_VENDAS
SELECT * FROM TST_BACKUP
```

Você verá que o seu valor será 11, que era o valor incluído após o backup de log das 17:00 PM:

POS
18
19
20
21
22
23
24

43) Você pode recuperar o backup através do **Management Studio**. Você já fez isto anteriormente, quando recuperou o backup da base de dados `SUCOS_VENDAS.BAK`. Repita aquele processo, mas usando o backup salvo no vídeo anterior.

44) Execute novamente todos os backups criados no vídeo anterior, ou seja:

```
-- 1:00 AM -- BACKUP FULL INICIAL
INSERT INTO TST_BACKUP (POS) VALUES (1)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH INIT

-- 4:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (2)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 6:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (3)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 8:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (4)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 9:00 AM - BACKUP DIFERENCIAL
INSERT INTO TST_BACKUP (POS) VALUES (5)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH DIFFERENTIAL
```

```
-- 10:00 AM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (6)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 12:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (7)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 14:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (8)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 14:00 PM - BACKUP DIFERENCIAL
INSERT INTO TST_BACKUP (POS) VALUES (9)
BACKUP DATABASE SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH DIFFERENTIAL

-- 15:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (10)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 17:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (11)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 19:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (12)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT

-- 21:00 PM - BACKUP LOG
INSERT INTO TST_BACKUP (POS) VALUES (13)
BACKUP LOG SUCOS_VENDAS TO DISK = 'C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK'
WITH NOINIT
```

45) Verifique o conteúdo da tabela de marcação:

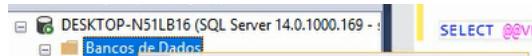
```
SELECT * FROM TST_BACKUP
```

POS	Value
7	7
8	8
9	9
10	10
11	11
12	12
13	13

46) Apague a base, executando:

```
USE MASTER
ALTER DATABASE SUCOS_VENDAS SET SINGLE_USER WITH ROLLBACK IMMEDIATE
DROP DATABASE SUCOS_VENDAS
```

47) Clique com o botão da direita do mouse sobre **Banco de Dados** e depois escolha **Restaurar Banco de Dados**:



48) Escolha o arquivo `C:\Temp\SQL\SUCOS_VENDAS_COMPLETO.BAK`:

49) Você verá, agora, a lista dos segmentos de backup. Se você vai recuperar o backup de 17:30 PM, tem que selecionar, nas opções ao lado, o arquivo que representa o último backup de log executado antes das 17:30 PM, que foi o de 17:00 PM:

50) Execute a recuperação e depois confira o valor da marcação na tabela:

```
USE SUCOS_VENDAS
SELECT * FROM TST_BACKUP
```

Você verá que o seu valor será 11, que era o valor incluído após o backup de log das 17:00 PM:

A screenshot of the SSMS results grid. The title bar says '99 %' and the tabs at the top are 'Resultados' and 'Mensagens'. The 'Resultados' tab is selected. The table has one column labeled 'POS' and seven rows of data. The last row, which contains the value '11', is highlighted with a blue selection bar.

POS
18
19
20
21
22
23
24