13

Mãos na massa: Juntando tabelas e consultas

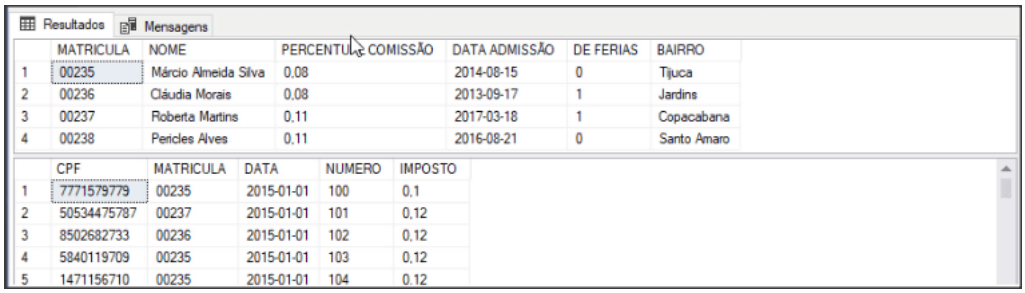
Chegou a hora de você executar o que foi visto na aula! Para isso, baixe aqui (https://s3.amazonaws.com/caelum-online-public/836-consultas-com-sql-server-2017/04/Downloads+-+Aula+4.zip) os arquivos necessários e siga os passos abaixo.

Usando o INNER JOIN

1) Abra o arquivo **Usando INNER JOIN.sql**.

2) Olhe as tabelas de vendedores e notas fiscais:

```
SELECT * FROM [TABELA DE VENDEDORES]
SELECT * FROM [NOTAS FISCAIS]
```



Resultados | Mensagens

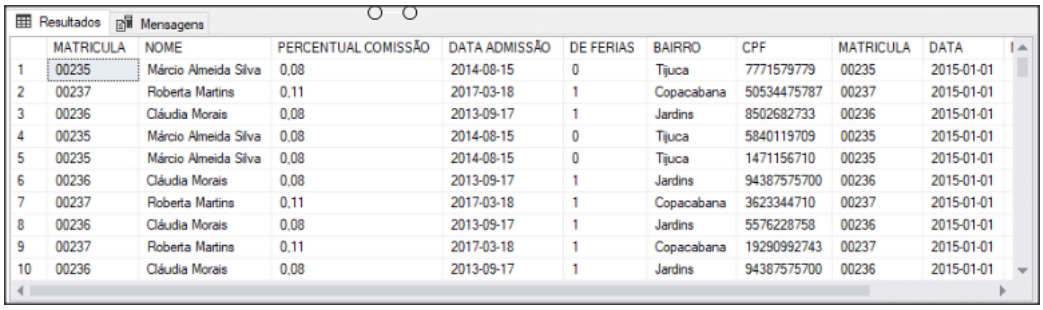
	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO
1	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca
2	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins
3	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana
4	00238	Pericles Alves	0,11	2016-08-21	0	Santo Amaro

	CPF	MATRICULA	DATA	NUMERO	IMPOSTO
1	7771579779	00235	2015-01-01	100	0,1
2	50534475787	00237	2015-01-01	101	0,12
3	8502682733	00236	2015-01-01	102	0,12
4	5840119709	00235	2015-01-01	103	0,12
5	1471156710	00235	2015-01-01	104	0,12

Há um campo em comum entre as duas tabelas, que é a `MATRICULA`. Por este campo é que você pode juntar as duas tabelas.

3) Execute:

```
SELECT * FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
```



Resultados | Mensagens

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01
6	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01
7	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	3623344710	00237	2015-01-01
8	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	5576228758	00236	2015-01-01
9	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	19290992743	00237	2015-01-01
10	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01

Neste resultado há todos os campos das duas tabelas.

4) Quando você faz a igualdade:

```
[TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
```

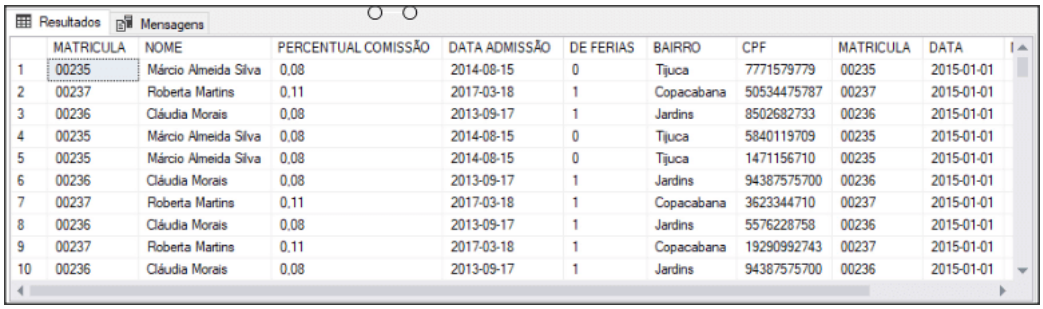
Tem que colocar o nome da tabela na frente do nome do campo. Isso porque se você usar:

```
MATRICULA = MATRICULA
```

O SQL Server não saberá de que tabela pertence os campos `MATRICULA`.

5) Crie um *alias*, que pode ser um nome ou uma letra, para colocar ao lado do nome da tabela. O comando ficará assim:

```
SELECT * FROM [TABELA DE VENDEDORES] A INNER JOIN
[NOTAS FISCAIS] B ON A.MATRICULA = B.MATRICULA
```

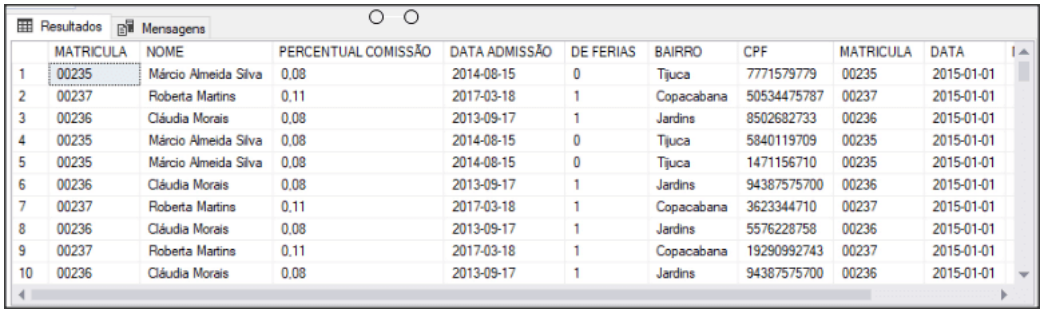


Resultados | Mensagens

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01
6	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01
7	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	3623344710	00237	2015-01-01
8	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	5576228758	00236	2015-01-01
9	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	19290992743	00237	2015-01-01
10	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01

6) Selecione os campos a serem exibidos no `JOIN`. Você pode aplicar tudo o que já foi visto até agora nesta seleção (filtros, `TOP`, `DISTINCT`, `GROUP BY`, `ORDER BY`, etc). Por exemplo, conte o número de notas fiscais por cada vendedor:

```
SELECT [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], COUNT(*)
FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
GROUP BY [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME]
```

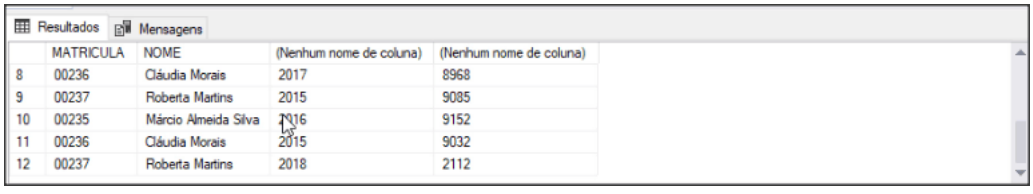


Resultados | Mensagens

	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01
5	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01
6	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01
7	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	3623344710	00237	2015-01-01
8	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	5576228758	00236	2015-01-01
9	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	19290992743	00237	2015-01-01
10	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	94387575700	00236	2015-01-01

7) Melhore a consulta e veja o número de notas fiscais, mas dentro de cada ano:

```
SELECT [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], YEAR(DATA),
COUNT(*) FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
GROUP BY [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], YEAR(DATA)
```



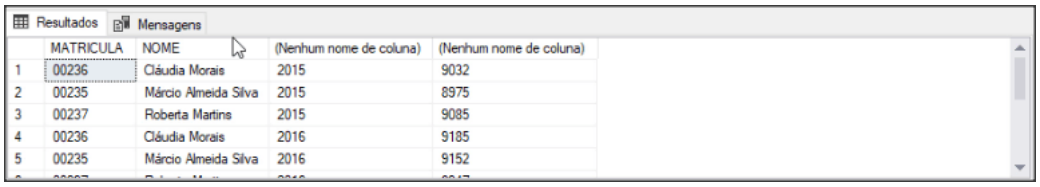
Resultados | Mensagens

	MATRICULA	NOME	(Nenhum nome de coluna)	(Nenhum nome de coluna)
8	00236	Cláudia Morais	2017	8968
9	00237	Roberta Martins	2015	9085
10	00235	Márcio Almeida Silva	2016	9152
11	00236	Cláudia Morais	2015	9032
12	00237	Roberta Martins	2018	2112

8) Nesta consulta final, ajuste a consulta acima, mostrando por ordem de data e vendedor:

```
SELECT [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], YEAR(DATA),
COUNT(*) FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA


GROUP BY [TABELA DE VENDEDORES].MATRICULA, [TABELA DE VENDEDORES].[NOME], YEAR(DATA)
ORDER BY YEAR(DATA), [TABELA DE VENDEDORES].[NOME]
```

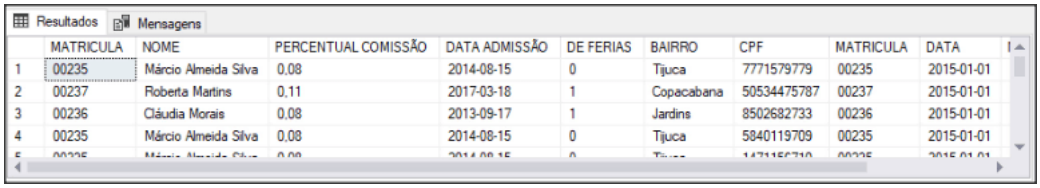


Resultados | Mensagens

	MATRICULA	NOME	(Nenhum nome de coluna)	(Nenhum nome de coluna)
1	00236	Cláudia Morais	2015	9032
2	00235	Márcio Almeida Silva	2015	8975
3	00237	Roberta Martins	2015	9085
4	00236	Cláudia Morais	2016	9185
5	00235	Márcio Almeida Silva	2016	9152

9) Existe uma forma antiga de fazer `JOIN`, que é considerá-lo como se fosse um filtro e declarar as tabelas que vão ser unidas após o `FROM`, com as mesmas separadas por vírgula:

```
SELECT * FROM [TABELA DE VENDEDORES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE VENDEDORES].MATRICULA = [NOTAS FISCAIS].MATRICULA
```



Resultados | Mensagens

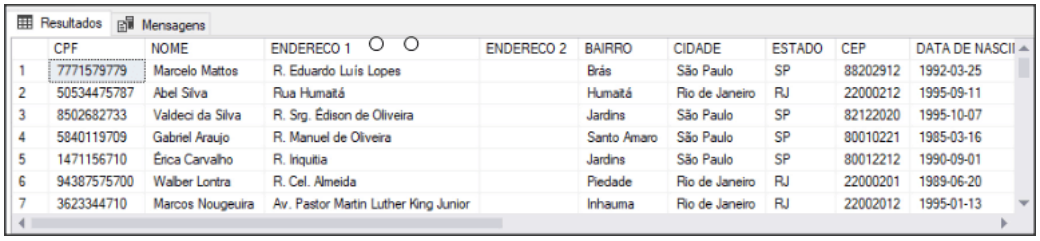
	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO	CPF	MATRICULA	DATA
1	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01
2	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana	50534475787	00237	2015-01-01
3	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins	8502682733	00236	2015-01-01
4	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01

Usando LEFT e RIGHT JOIN

10) Abra o arquivo **Usando LEFT e RIGHT JOIN.sql**.

11) Junte a tabelas de clientes com a tabelas de notas fiscais pelo campo `CPF`, que é o campo em comum das duas:

```
SELECT * FROM [TABELA DE CLIENTES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
```

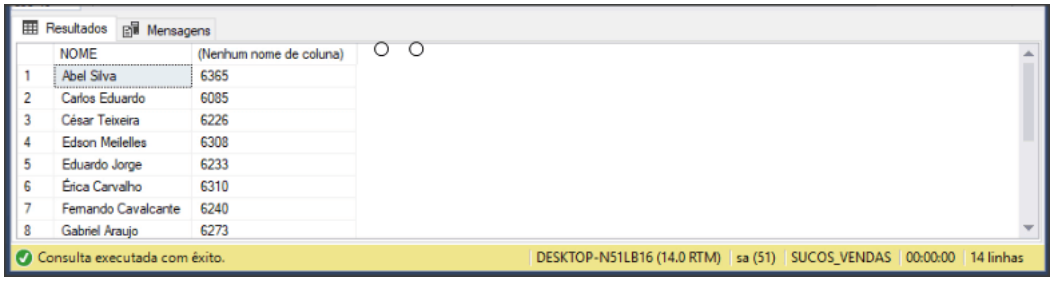


Resultados | Mensagens

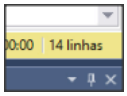
	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCI
1	7771579779	Marcelo Mattos	R. Eduardo Luís Lopes		Brás	São Paulo	SP	88202912	1992-03-25
2	50534475787	Abel Silva	Rua Humaitá		Humaitá	Rio de Janeiro	RJ	22000212	1995-09-11
3	8502682733	Valdeci da Silva	R. Srg. Édison de Oliveira		Jardins	São Paulo	SP	82122020	1995-10-07
4	5840119709	Gabriel Araujo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16
5	1471156710	Érica Carvalho	R. Iriquitia		Jardins	São Paulo	SP	80012212	1990-09-01
6	94387575700	Walber Lontra	R. Cel. Almeida		Piedade	Rio de Janeiro	RJ	22000201	1989-06-20
7	3623344710	Marcos Nougeuira	Av. Pastor Martin Luther King Junior		Inhauma	Rio de Janeiro	RJ	22002012	1995-01-13

12) Aplicando o `COUNT(*)` com `GROUP BY`, conte o número de notas fiscais por cada cliente:

```
SELECT [TABELA DE CLIENTES].[NOME], COUNT(*) FROM [TABELA DE CLIENTES] INNER JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
GROUP BY [TABELA DE CLIENTES].[NOME]
```

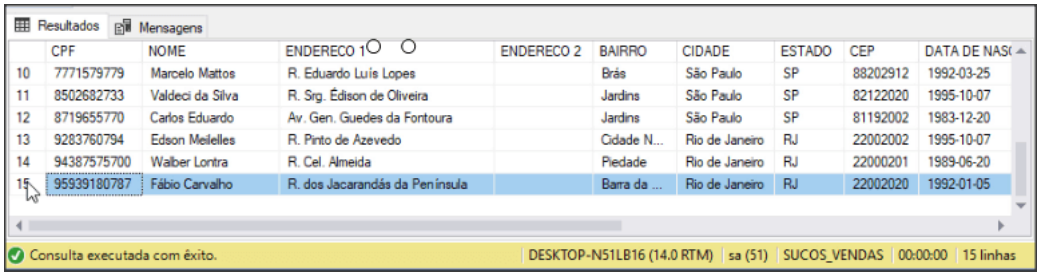


Note que esta consulta retornou 14 linhas:

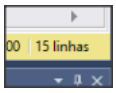


13) Verifique quantos clientes há na tabela de clientes:

```
SELECT * FROM [TABELA DE CLIENTES]
```



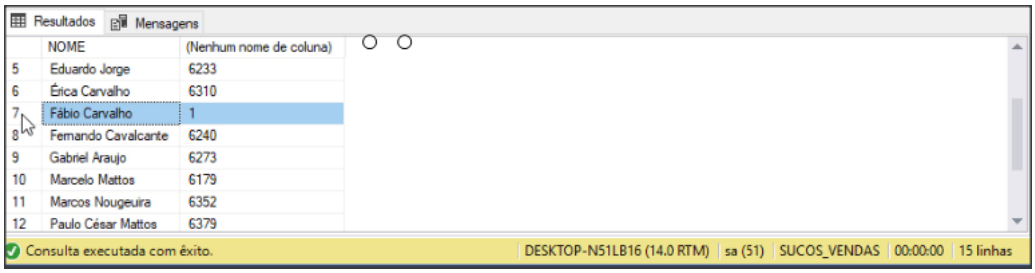
Esta consulta retorna 15 linhas:



Isto significa que, quando você vê a tabela de clientes, há 15 linhas, mas quando agrupa a consulta, fazendo `JOIN` com a tabela de notas fiscais, há 14 clientes. Logo, existe um cliente que faz parte do cadastro e que nunca fez uma compra. Pelo `LEFT JOIN`, você pode podemos descobrir este cliente.

14) Para descobrir o cliente que nunca comprou, faça:

```
SELECT [TABELA DE CLIENTES].[NOME], COUNT(*) FROM [TABELA DE CLIENTES] LEFT JOIN
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
GROUP BY [TABELA DE CLIENTES].[NOME]
```

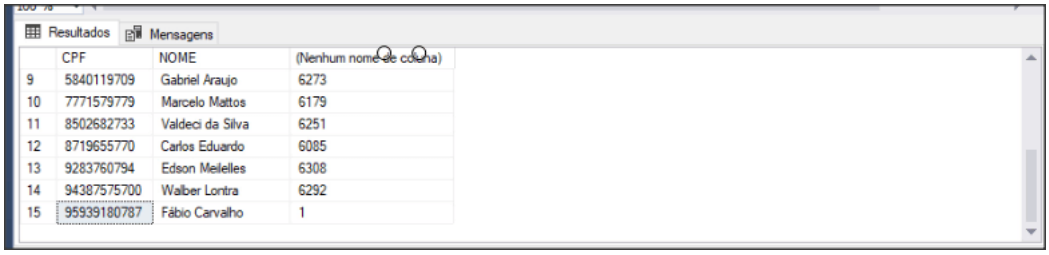


O cliente que tem valor de nota fiscal igual a `1` é o cliente que nunca comprou. O valor `1` aparece porque o `COUNT(*)` faz uma contagem de registros e este cliente aparece na consulta, apesar de não ter nenhuma nota. Por causa do `LEFT JOIN`, ele aparece apenas uma vez e por isso o `COUNT(*)` escreve para ele o valor `1`.

15) Para ter certeza deste resultado, coloque o CPF na consulta:

```
SELECT [TABELA DE CLIENTES].[CPF], [TABELA DE CLIENTES].[NOME], COUNT(*) FROM [TABELA DE CLIENTES] l
[NOTAS FISCAIS] ON [TABELA DE CLIENTES].CPF = [NOTAS FISCAIS].CPF
GROUP BY  [TABELA DE CLIENTES].[CPF], [TABELA DE CLIENTES].[NOME]
```

Você terá:



	CPF	NOME	(Nenhum nome de coluna)
9	5840119709	Gabriel Araujo	6273
10	7771579779	Marcelo Mattos	6179
11	8502682733	Valdeci da Silva	6251
12	8719655770	Carlos Eduardo	6085
13	9283760794	Edson Meilelles	6308
14	94387575700	Walber Lontra	6292
15	95939180787	Fábio Carvalho	1

Pegando o CPF que aparece na consulta acima, você pode ver se há alguma nota fiscal associada a ele na tabela de notas fiscais:

```
SELECT * FROM [NOTAS FISCAIS] WHERE CPF = '95939180787'
```

Como esperado, resultado vazio:

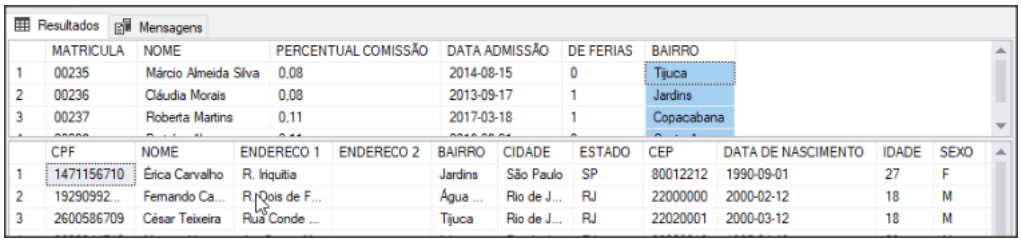
CPF	MATRICULA	DATA	NUMERO	IMPOSTO

Usando FULL e CROSS JOIN

16) Abra o aquivo **Usando FULL e CROSS JOIN.sql**.

17) Veja a tabela de vendedores e de clientes:

```
SELECT * FROM [TABELA DE VENDEDORES]
SELECT * FROM [TABELA DE CLIENTES]
```



	MATRICULA	NOME	PERCENTUAL COMISSÃO	DATA ADMISSÃO	DE FERIAS	BAIRRO
1	00235	Márcio Almeida Silva	0,08	2014-08-15	0	Tijuca
2	00236	Cláudia Morais	0,08	2013-09-17	1	Jardins
3	00237	Roberta Martins	0,11	2017-03-18	1	Copacabana

	CPF	NOME	ENDERECO 1	ENDERECO 2	BAIRRO	CIDADE	ESTADO	CEP	DATA DE NASCIMENTO	IDADE	SEXO
1	1471156710	Érica Carvalho	R. Iriquitia		Jardins	São Paulo	SP	80012212	1990-09-01	27	F
2	19290992...	Fernando Ca...	R. Dois de F...		Água ...	Rio de J...	RJ	22000000	2000-02-12	18	M
3	2600586709	César Teixeira	Rua Conde ...		Tijuca	Rio de J...	RJ	22020001	2000-03-12	18	M

Resultado: 4 vendedores e 15 clientes. O campo em comum entre eles é o `BAIRRO`.

18) Faça o `INNER JOIN` entre a tabela de clientes e de vendedores:

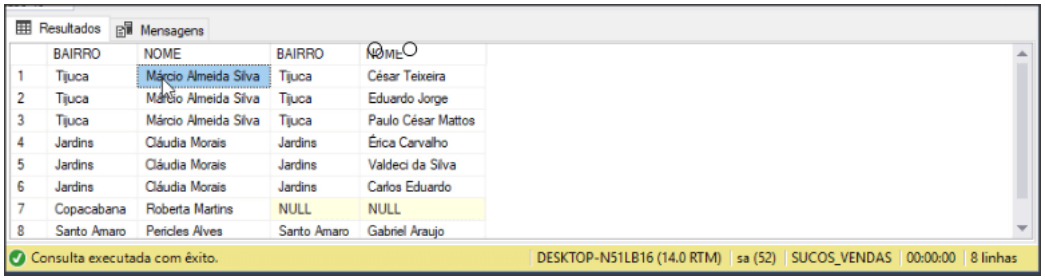
```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE CLIENTES].BAIRRO FROM
[TABELA DE VENDEDORES] INNER JOIN [TABELA DE CLIENTES] ON
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```

Resultados | Mensagens

	BAIRRO	BAIRRO
1	Jardins	Jardins
2	Tijuca	Tijuca
3	Tijuca	Tijuca
4	Tijuca	Tijuca
5	Santo Amaro	Santo Amaro
6	Jardins	Jardins
7	Jardins	Jardins

19) Como o resultado foi de 7 linhas, significa que existem vendedores com bairros que não estão na tabela de clientes e vice-versa. O `LEFT JOIN` pode fazer com que você possa descobrir estas não correspondências:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME],
[TABELA DE CLIENTES].BAIRRO, [TABELA DE CLIENTES].[NOME] FROM
[TABELA DE VENDEDORES] LEFT JOIN [TABELA DE CLIENTES] ON
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```



Resultados | Mensagens

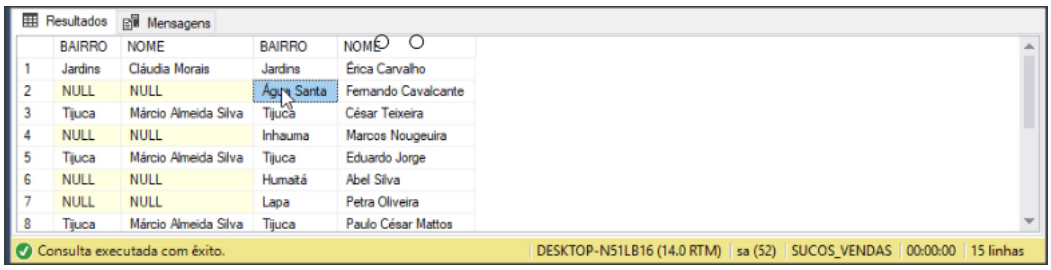
	BAIRRO	NOME	BAIRRO	NOME
1	Tijuca	Márcio Almeida Silva	Tijuca	César Teixeira
2	Tijuca	Márcio Almeida Silva	Tijuca	Eduardo Jorge
3	Tijuca	Márcio Almeida Silva	Tijuca	Paulo César Mattos
4	Jardins	Cláudia Morais	Jardins	Érica Carvalho
5	Jardins	Cláudia Morais	Jardins	Valdeci da Silva
6	Jardins	Cláudia Morais	Jardins	Carlos Eduardo
7	Copacabana	Roberta Martins	NULL	NULL
8	Santo Amaro	Pericles Alves	Santo Amaro	Gabriel Araujo

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS_VENDAS | 00:00:00 | 8 linhas

Resultado: 8 linhas

20) Como você ainda não obteve o resultado da maior tabela, significa que existem também vendedores com bairros que não existem na tabela de clientes. Para isso, use o `RIGHT JOIN` para descobrir isto:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME],
[TABELA DE CLIENTES].BAIRRO, [TABELA DE CLIENTES].[NOME] FROM
[TABELA DE VENDEDORES] RIGHT JOIN [TABELA DE CLIENTES] ON
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```



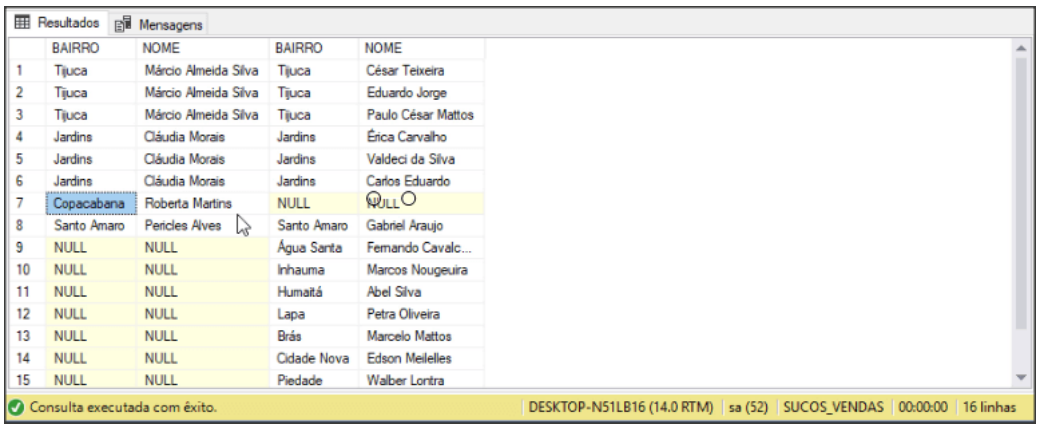
Resultados | Mensagens

	BAIRRO	NOME	BAIRRO	NOME
1	Jardins	Cláudia Morais	Jardins	Érica Carvalho
2	NULL	NULL	Água Santa	Fernando Cavalcante
3	Tijuca	Márcio Almeida Silva	Tijuca	César Teixeira
4	NULL	NULL	Inhaúma	Marcos Nougueira
5	Tijuca	Márcio Almeida Silva	Tijuca	Eduardo Jorge
6	NULL	NULL	Humaitá	Abel Silva
7	NULL	NULL	Lapa	Petra Oliveira
8	Tijuca	Márcio Almeida Silva	Tijuca	Paulo César Mattos

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS_VENDAS | 00:00:00 | 15 linhas

21) Para descobrir, ao mesmo tempo, o cliente que não tem bairro na tabela de vendedores e o vendedor que não tem bairro na tabela de clientes, use o `FULL JOIN` :

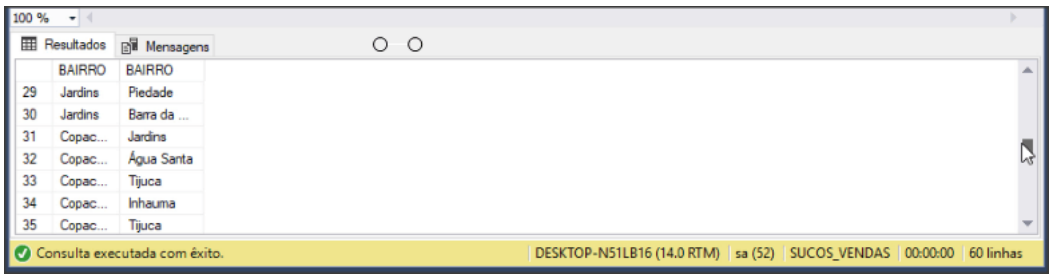
```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME],
[TABELA DE CLIENTES].BAIRRO, [TABELA DE CLIENTES].[NOME] FROM
[TABELA DE VENDEDORES] FULL JOIN [TABELA DE CLIENTES] ON
[TABELA DE VENDEDORES].BAIRRO = [TABELA DE CLIENTES].BAIRRO
```



	BAIRRO	NOME	BAIRRO	NOME
1	Tijuca	Márcio Almeida Silva	Tijuca	César Teixeira
2	Tijuca	Márcio Almeida Silva	Tijuca	Eduardo Jorge
3	Tijuca	Márcio Almeida Silva	Tijuca	Paulo César Mattos
4	Jardins	Cláudia Morais	Jardins	Érica Carvalho
5	Jardins	Cláudia Morais	Jardins	Valdeci da Silva
6	Jardins	Cláudia Morais	Jardins	Carlos Eduardo
7	Copacabana	Roberta Martins	NULL	NULL
8	Santo Amaro	Péricles Alves	Santo Amaro	Gabriel Araujo
9	NULL	NULL	Água Santa	Fernando Cavalc...
10	NULL	NULL	Inhauma	Marcos Nougeuira
11	NULL	NULL	Humaitá	Abel Silva
12	NULL	NULL	Lapa	Petra Oliveira
13	NULL	NULL	Brás	Marcelo Mattos
14	NULL	NULL	Cidade Nova	Edson Meilelles
15	NULL	NULL	Piedade	Walber Lontra

22) O `CROSS JOIN` irá possibilitar a análise combinatória da tabela de vendedores e clientes:

```
SELECT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE CLIENTES].BAIRRO FROM
[TABELA DE VENDEDORES] CROSS JOIN [TABELA DE CLIENTES]
```



	BAIRRO	BAIRRO
29	Jardins	Piedade
30	Jardins	Barra da ...
31	Copac...	Jardins
32	Copac...	Água Santa
33	Copac...	Tijuca
34	Copac...	Inhauma
35	Copac...	Tijuca

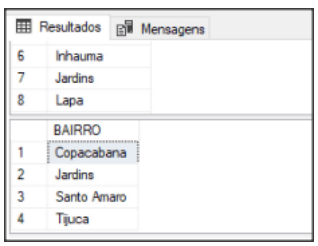
O resultado foi justamente 60 linhas, que é são os 15 clientes vezes os 4 vendedores.

Juntando consultas

23) Abra o arquivo **Juntando consultas.sql**.

24) Veja a lista de bairros dos clientes e dos vendedores:

```
SELECT DISTINCT [TABELA DE CLIENTES].BAIRRO FROM [TABELA DE CLIENTES]
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO FROM [TABELA DE VENDEDORES]
```

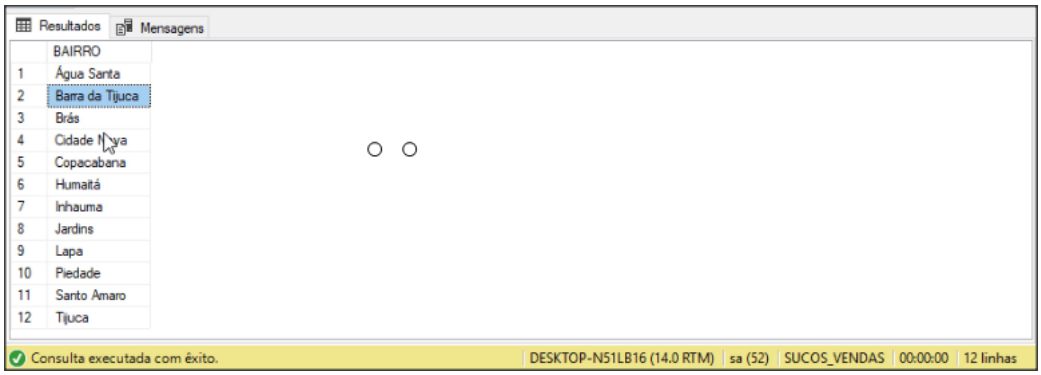


6	Inhauma
7	Jardins
8	Lapa

	BAIRRO
1	Copacabana
2	Jardins
3	Santo Amaro
4	Tijuca

25) Use o `UNION` , juntando essas duas consultas:

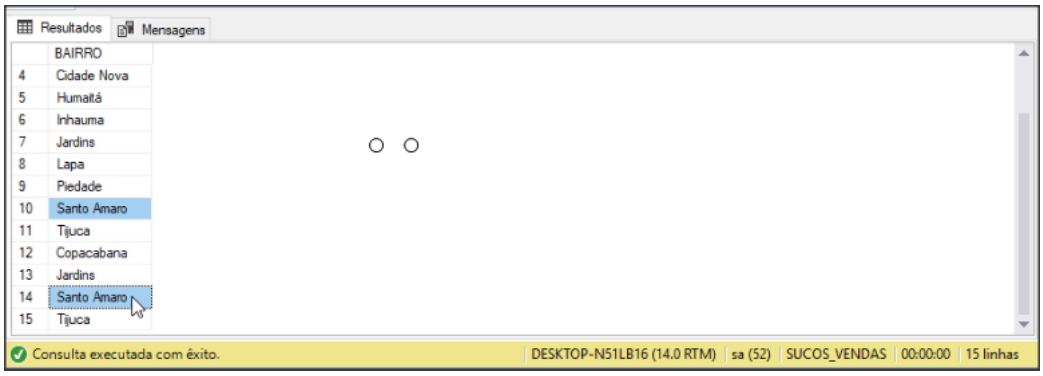
```
SELECT DISTINCT [TABELA DE CLIENTES].BAIRRO FROM [TABELA DE CLIENTES]
UNION
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO FROM [TABELA DE VENDEDORES]
```



Com o **UNION** , você tem 12 bairros sem que nenhum se repita.

26) Execute agora o **UNION ALL** . Teremos:

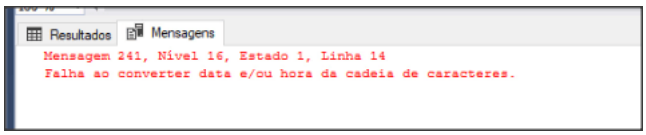
```
SELECT DISTINCT [TABELA DE CLIENTES].BAIRRO FROM [TABELA DE CLIENTES]
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO FROM [TABELA DE VENDEDORES]
```



Você terá 15 linhas como resultado, porque os bairros em comum aparecem duplicados.

27) Os campos de um **UNION** devem ter a mesma correspondência de tipo. Veja a consulta abaixo:

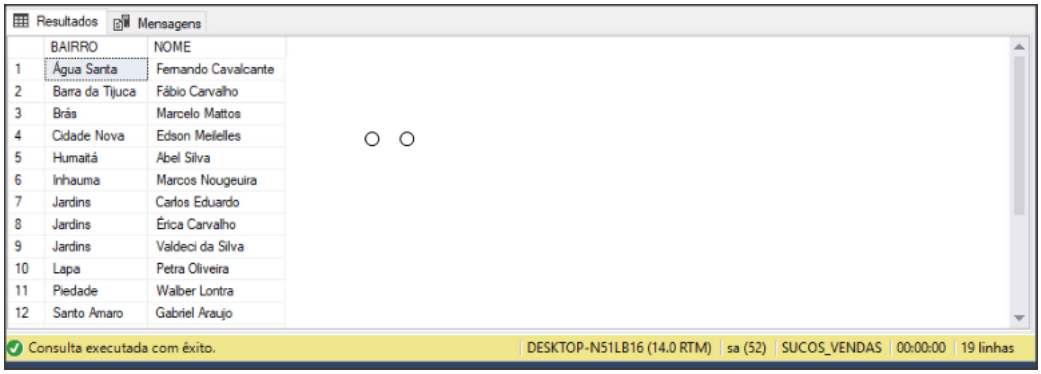
```
SELECT DISTINCT [TABELA DE CLIENTES].[BAIRRO], [TABELA DE CLIENTES].[DATA DE NASCIMENTO] FROM [TABEI
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME] FROM [TABELA DE VENDEDO
```



Você terá um erro porque o segundo campo das duas consultas que estão sendo unidas pelo **UNION** são de tipos diferentes.

28) Execute então:

```
SELECT DISTINCT [TABELA DE CLIENTES].[BAIRRO], [TABELA DE CLIENTES].[NOME] FROM [TABELA DE CLIENTES]
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME] FROM [TABELA DE VENDEDO
```

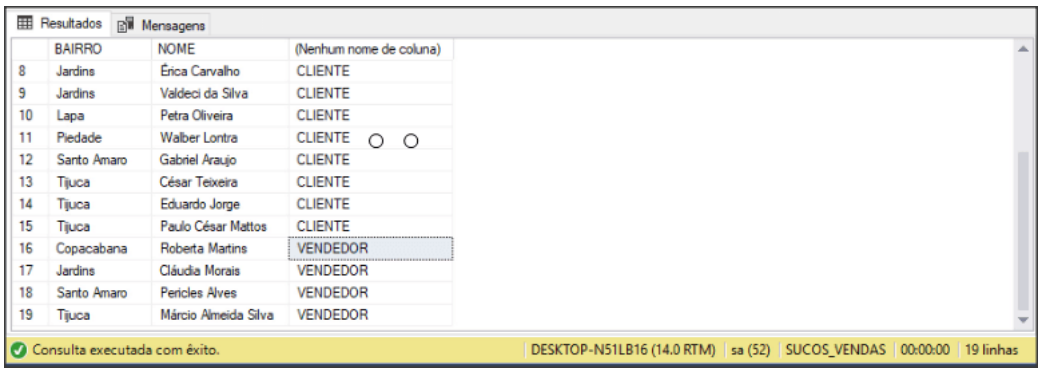


	BAIRRO	NOME
1	Água Santa	Fernando Cavalcante
2	Barra da Tijuca	Fábio Carvalho
3	Brás	Marcelo Mattos
4	Cidade Nova	Edson Mellelles
5	Humaitá	Abel Silva
6	Inhauma	Marcos Nougeuira
7	Jardins	Carlos Eduardo
8	Jardins	Érica Carvalho
9	Jardins	Valdeci da Silva
10	Lapa	Petra Oliveira
11	Piedade	Walber Lontra
12	Santo Amaro	Gabriel Araujo

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS_VENDAS | 00:00:00 | 19 linhas

29) O que você pode fazer, em muitos dos casos, é criar uma coluna que identifique a origem do campo quando é efetuado o **UNION** :

```
SELECT DISTINCT [TABELA DE CLIENTES].[BAIRRO], [TABELA DE CLIENTES].[NOME], 'CLIENTE' FROM [TABELA D
UNION ALL
SELECT DISTINCT [TABELA DE VENDEDORES].BAIRRO, [TABELA DE VENDEDORES].[NOME], 'VENDEDOR' FROM [TABEL
```



	BAIRRO	NOME	(Nenhum nome de coluna)
8	Jardins	Érica Carvalho	CLIENTE
9	Jardins	Valdeci da Silva	CLIENTE
10	Lapa	Petra Oliveira	CLIENTE
11	Piedade	Walber Lontra	CLIENTE
12	Santo Amaro	Gabriel Araujo	CLIENTE
13	Tijuca	César Teixeira	CLIENTE
14	Tijuca	Eduardo Jorge	CLIENTE
15	Tijuca	Paulo César Mattos	CLIENTE
16	Copacabana	Roberta Martins	VENDEDOR
17	Jardins	Cláudia Morais	VENDEDOR
18	Santo Amaro	Pericles Alves	VENDEDOR
19	Tijuca	Márcio Almeida Silva	VENDEDOR

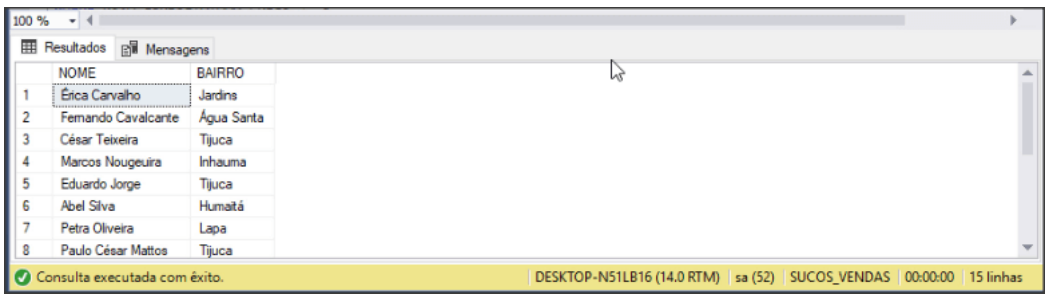
Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS_VENDAS | 00:00:00 | 19 linhas

Sub-consultas

30) Abra o arquivo **Sub consultas.sql**.

31) Veja a consulta de clientes e seus respectivos bairros:

```
SELECT NOME, BAIRRO FROM [TABELA DE CLIENTES]
```



	NOME	BAIRRO
1	Érica Carvalho	Jardins
2	Fernando Cavalcante	Água Santa
3	César Teixeira	Tijuca
4	Marcos Nougeuira	Inhauma
5	Eduardo Jorge	Tijuca
6	Abel Silva	Humaitá
7	Petra Oliveira	Lapa
8	Paulo César Mattos	Tijuca

Consulta executada com êxito. | DESKTOP-N51LB16 (14.0 RTM) | sa (52) | SUCOS_VENDAS | 00:00:00 | 15 linhas

32) Liste somente os clientes que possuam os mesmos bairros que os vendedores. Como você não sabe de antemão quais são os bairros dos vendedores, use a sub-consulta como condição de filtro:

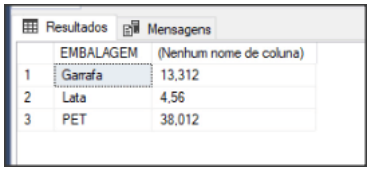
```
SELECT NOME, BAIRRO FROM [TABELA DE CLIENTES]
WHERE BAIRRO IN (
SELECT BAIRRO FROM [TABELA DE VENDEDORES])
```


Resultados Mensagens

	NOME	BAIRRO
1	Érica Carvalho	Jardins
2	César Teixeira	Tijuca
3	Eduardo Jorge	Tijuca
4	Paulo César Mattos	Tijuca
5	Gabriel Araujo	Santo Amaro
6	Valdeci da Silva	Jardins
7	Carlos Eduardo	Jardins

33) Veja o máximo dos preços por embalagem:

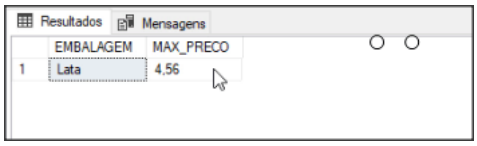
```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) FROM [TABELA DE PRODUTOS] GROUP BY EMBALAGEM
```


Resultados Mensagens

	EMBALAGEM	(Nenhum nome de coluna)
1	Garrafa	13,312
2	Lata	4,56
3	PET	38,012

34) Aplique um filtro sobre esta consulta, usando ela como uma sub-consulta no `FROM` de outra consulta:

```
SELECT NOVA_CONSULTA.EMBALAGEM, NOVA_CONSULTA.MAX_PRECO
FROM (SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE PRODUTOS] GROUP BY EMBALA
NOVA_CONSULTA
WHERE NOVA_CONSULTA.MAX_PRECO <= 5
```


Resultados Mensagens

	EMBALAGEM	MAX_PRECO
1	Lata	4,56

O resultado acima acaba fazendo o mesmo que o `HAVING` .

Visão

35) Abra o arquivo **Visões.sql**.

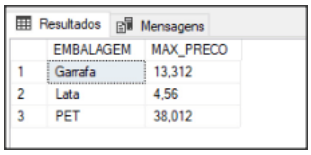
36) Note que na consulta abaixo:

```
SELECT NOVA_CONSULTA.EMBALAGEM, NOVA_CONSULTA.MAX_PRECO
FROM (SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE PRODUTOS] GROUP BY EMBALA
NOVA_CONSULTA
WHERE NOVA_CONSULTA.MAX_PRECO <= 5
```

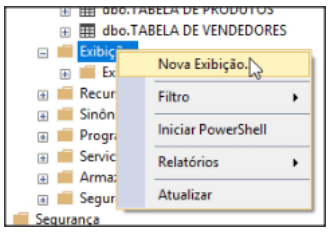
Há uma sub-consulta:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE PRODUTOS] GROUP BY EMBALAGEM
```

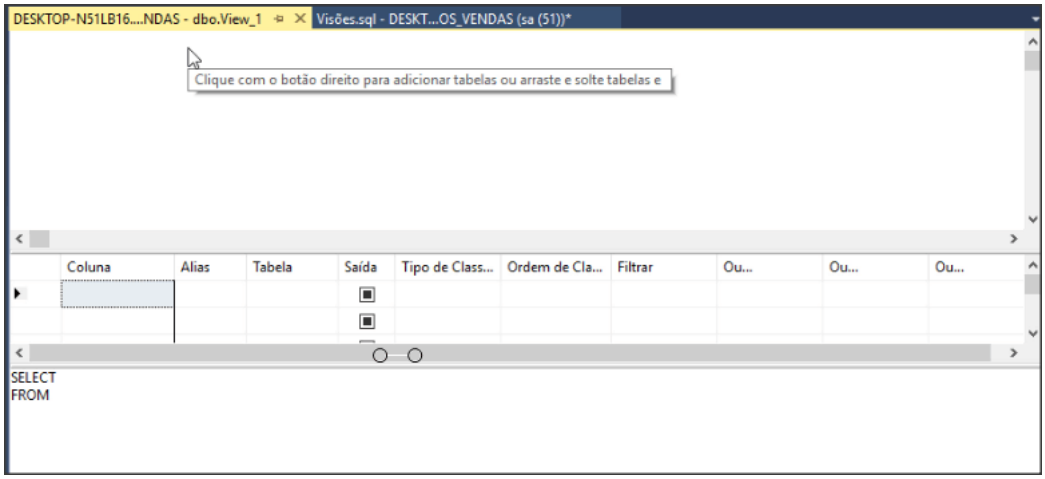
Executando esta sub-consulta, você verá:



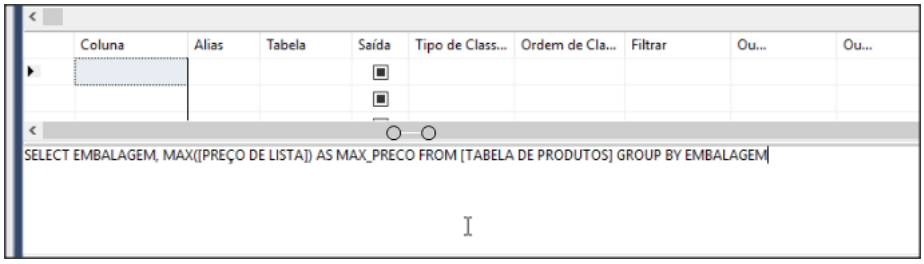
37) Expanda a pasta **Exibições** do banco de dados **SUCOS_VENDAS**. Clique com o botão da direita do mouse sobre este item e escolha **Nova Exibição**:



38) Serão mostradas as tabelas. Aqui, você pode escolher as tabelas que farão parte das *views*. Mas, no caso, já há a consulta SQL pronta. Logo, clique em **OK** sem selecionar tabela alguma:

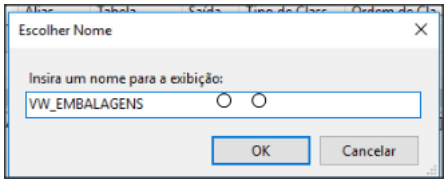


39) onde há escrito `SELECT FROM` , cole a sub-consulta que será usada para a criação da *view*:

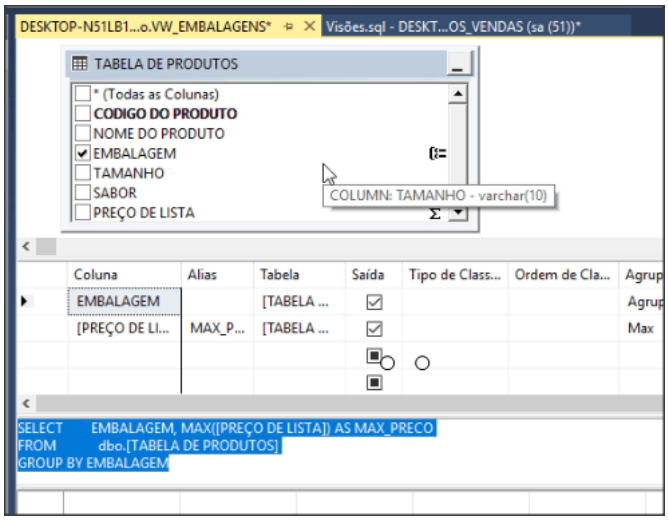


40) Clique no ícone **Salvar** na barra de ferramentas principal:

41) Crie um nome para esta visão (**VW_EMBALAGENS**):

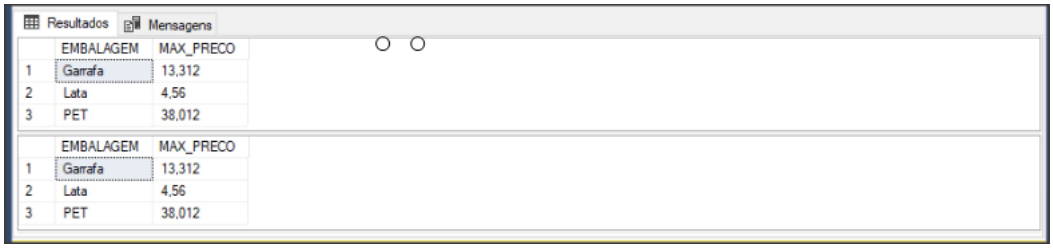


42) Clique em **OK** e você terá a representação gráfica da visão:



43) Agora, use a visão na consulta como se fosse uma tabela. Veja abaixo a seleção original e a visão:

```
SELECT EMBALAGEM, MAX([PREÇO DE LISTA]) AS MAX_PRECO FROM [TABELA DE PRODUTOS] GROUP BY EMBALAGEM
SELECT * FROM [dbo].[VW_EMBALAGENS]
```



44) Você pode usar a visão como se fosse uma tabela na consulta que, antes, foi usada a sub-consulta:

```
SELECT NOVA_CONSULTA.EMBALAGEM, NOVA_CONSULTA.MAX_PRECO
FROM VW_EMBALAGENS NOVA_CONSULTA
WHERE NOVA_CONSULTA.MAX_PRECO <= 5
```