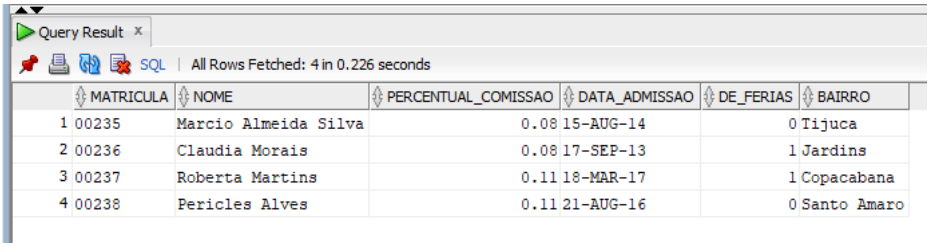
14

Consolidando o seu conhecimento

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

1) Veja o conteúdo das tabelas `TABELA_DE_VENDEDORES` e `NOTAS_FISCAIS`, digitando os comandos abaixo:

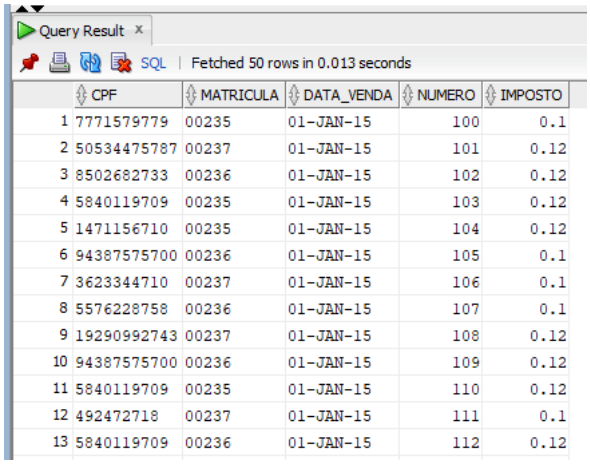
```
SELECT * FROM TABELA_DE_VENDEDORES;
```



Query Result | SQL | All Rows Fetched: 4 in 0.226 seconds

	MATRICULA	NOME	PERCENTUAL_COMISSAO	DATA_ADMISSAO	DE_FERIAS	BAIRRO
1	00235	Marcio Almeida Silva	0.08	15-AUG-14	0	Tijuca
2	00236	Claudia Morais	0.08	17-SEP-13	1	Jardins
3	00237	Roberta Martins	0.11	18-MAR-17	1	Copacabana
4	00238	Pericles Alves	0.11	21-AUG-16	0	Santo Amaro

```
SELECT * FROM NOTAS_FISCAIS;
```

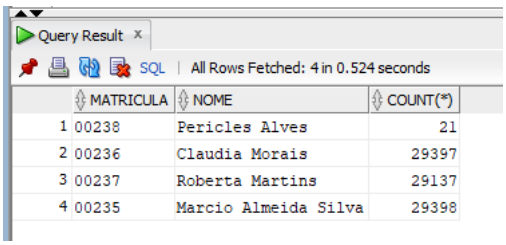


Query Result | SQL | Fetched 50 rows in 0.013 seconds

	CPF	MATRICULA	DATA_VENDA	NUMERO	IMPOSTO
1	7771579779	00235	01-JAN-15	100	0.1
2	50534475787	00237	01-JAN-15	101	0.12
3	8502682733	00236	01-JAN-15	102	0.12
4	5840119709	00235	01-JAN-15	103	0.12
5	1471156710	00235	01-JAN-15	104	0.12
6	94387575700	00236	01-JAN-15	105	0.1
7	3623344710	00237	01-JAN-15	106	0.1
8	5576228758	00236	01-JAN-15	107	0.1
9	19290992743	00237	01-JAN-15	108	0.12
10	94387575700	00236	01-JAN-15	109	0.12
11	5840119709	00235	01-JAN-15	110	0.12
12	492472718	00237	01-JAN-15	111	0.1
13	5840119709	00236	01-JAN-15	112	0.12

2) Você pode conectar essas duas tabelas pelo campo em comum, que é `MATRICULA`. Digite:

```
SELECT A.MATRICULA, B.NOME, COUNT(*) FROM NOTAS_FISCAIS A
INNER JOIN TABELA_DE_VENDEDORES B
ON A.MATRICULA = B.MATRICULA
GROUP BY A.MATRICULA, B.NOME
```

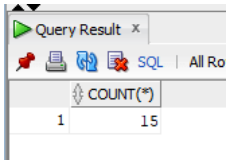


Query Result | SQL | All Rows Fetched: 4 in 0.524 seconds

	MATRICULA	NOME	COUNT(*)
1	00238	Pericles Alves	21
2	00236	Claudia Morais	29397
3	00237	Roberta Martins	29137
4	00235	Marcio Almeida Silva	29398

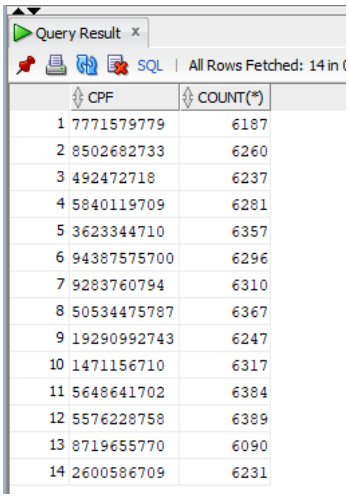
3) Nem sempre todas as linhas podem ser conectadas. Existem outros tipos de `JOIN` que permite identificar quem não pode ser conectado. Veja a consulta abaixo, que mostra que há 15 clientes:

```
SELECT COUNT(*) FROM TABELA_DE_CLIENTES;
```



4) Verifique quantos clientes realizaram compras na empresa de suco de frutas:

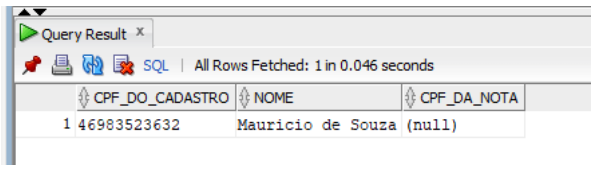
```
SELECT CPF, COUNT(*) FROM NOTAS_FISCAIS GROUP BY CPF;
```



Se você contar verá que, na consulta acima, há 14 linhas. Existe um cliente que está no cadastro mas não teve nota fiscal emitida.

5) Você pode usar o `LEFT JOIN`. Por exemplo:

```
SELECT DISTINCT A.CPF AS CPF_DO_CADASTRO, A.NOME, B.CPF AS CPF_DA_NOTA
FROM TABELA_DE_CLIENTES A LEFT JOIN NOTAS_FISCAIS B
ON A.CPF = B.CPF
WHERE B.CPF IS NULL;
```



O cliente que possui o CPF vindo da tabela de notas com o valor nulo é o cliente que nunca emitiu nota fiscal.

6) Há outros tipos de `JOIN`:

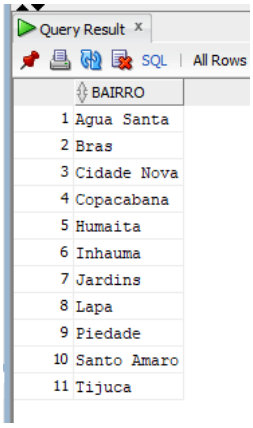
```
SELECT TABELA_DE_VENDEDORES.NOME AS NOME_VENDEDOR,
TABELA_DE_VENDEDORES.BAIRRO AS BAIRRO_VENDEDOR,
TABELA_DE_CLIENTES.NOME AS NOME_CLIENTE,
TABELA_DE_CLIENTES.BAIRRO AS BAIRRO_CLIENTE
FROM TABELA_DE_VENDEDORES
FULL JOIN TABELA_DE_CLIENTES
ON TABELA_DE_VENDEDORES.BAIRRO = TABELA_DE_CLIENTES.BAIRRO;
```


Query Result | SQL | All Rows Fetched: 16 in 0.028 seconds

	NOME_VENDEDOR	BAIRRO_VENDEDOR	NOME_CLIENTE	BAIRRO_CLIENTE
1	Claudia Morais	Jardins	Erica Carvalho	Jardins
2	(null)	(null)	Fernando Cavalcante	Agua Santa
3	Marcio Almeida Silva	Tijuca	Cesar Teixeira	Tijuca
4	(null)	(null)	Marcos Nogueira	Inhauma
5	Marcio Almeida Silva	Tijuca	Eduardo Jorge	Tijuca
6	(null)	(null)	Abel Silva	Humaita
7	(null)	(null)	Petra Oliveira	Lapa
8	Marcio Almeida Silva	Tijuca	Paulo Cesar Mattos	Tijuca
9	Pericles Alves	Santo Amaro	Gabriel Araujo	Santo Amaro
10	(null)	(null)	Marcelo Mattos	Bras
11	Claudia Morais	Jardins	Valdeci da Silva	Jardins
12	Claudia Morais	Jardins	Carlos Eduardo	Jardins
13	(null)	(null)	Edson Meilelles	Cidade Nova
14	(null)	(null)	Walber Lontra	Piedade
15	Claudia Morais	Jardins	Mauricio de Souza	Jardins

7) Você pode juntar duas ou mais consultas, desde que os campos selecionados sejam os mesmos. Por exemplo:

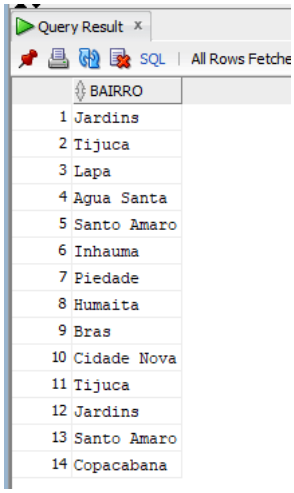
```
SELECT DISTINCT BAIRRO FROM TABELA_DE_CLIENTES
UNION
SELECT DISTINCT BAIRRO FROM TABELA_DE_VENDEDORES;
```


Query Result | SQL | All Rows

	BAIRRO
1	Agua Santa
2	Bras
3	Cidade Nova
4	Copacabana
5	Humaita
6	Inhauma
7	Jardins
8	Lapa
9	Piedade
10	Santo Amaro
11	Tijuca

8) O `UNION ALL` não faz a seleção com um `DISTINCT`. As linhas se repetem se existirem em ambas as tabelas. Por exemplo:

```
SELECT DISTINCT BAIRRO FROM TABELA_DE_CLIENTES
UNION ALL
SELECT DISTINCT BAIRRO FROM TABELA_DE_VENDEDORES;
```

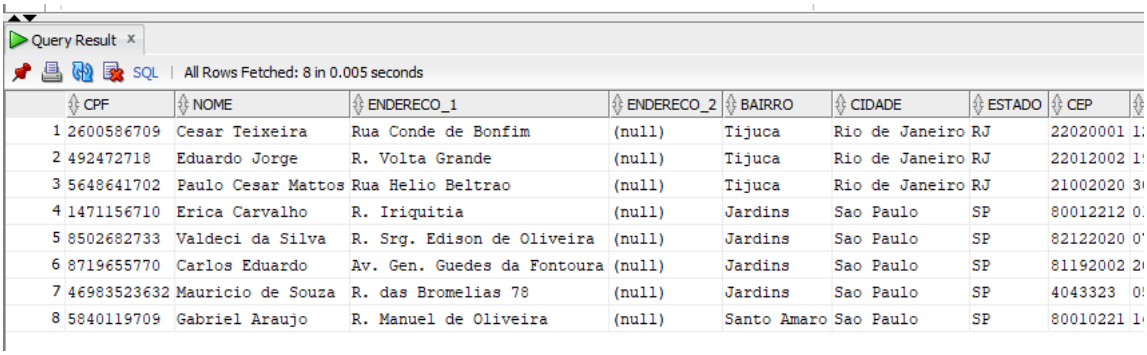

Query Result | SQL | All Rows Fetche

	BAIRRO
1	Jardins
2	Tijuca
3	Lapa
4	Agua Santa
5	Santo Amaro
6	Inhauma
7	Piedade
8	Humaita
9	Bras
10	Cidade Nova
11	Tijuca
12	Jardins
13	Santo Amaro
14	Copacabana

Veja que Santo Amaro aparece duas vezes. Uma vindo da tabela de clientes e outra da tabela de produtos.

9) As subconsultas permitem que seleções possam ser feitas usando como critérios outras seleções. Por exemplo:

```
SELECT * FROM TABELA_DE_CLIENTES WHERE BAIRRO
IN (SELECT DISTINCT BAIRRO FROM TABELA_DE_VENDEDORES);
```

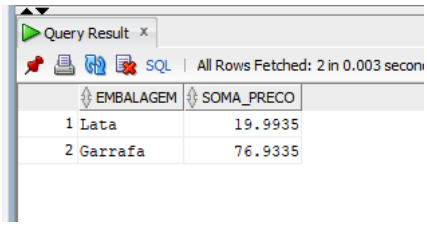


Query Result | SQL | All Rows Fetched: 8 in 0.005 seconds

	CPF	NOME	ENDERECO_1	ENDERECO_2	BAIRRO	CIDADE	ESTADO	CEP
1	2600586709	Cesar Teixeira	Rua Conde de Bonfim	(null)	Tijuca	Rio de Janeiro	RJ	22020001
2	492472718	Eduardo Jorge	R. Volta Grande	(null)	Tijuca	Rio de Janeiro	RJ	22012002
3	5648641702	Paulo Cesar Mattos	Rua Helio Beltrao	(null)	Tijuca	Rio de Janeiro	RJ	21002020
4	1471156710	Erica Carvalho	R. Iriquitia	(null)	Jardins	Sao Paulo	SP	80012212
5	8502682733	Valdeci da Silva	R. Srg. Edison de Oliveira	(null)	Jardins	Sao Paulo	SP	82122020
6	8719655770	Carlos Eduardo	Av. Gen. Guedes da Fontoura	(null)	Jardins	Sao Paulo	SP	81192002
7	46983523632	Mauricio de Souza	R. das Bromelias 78	(null)	Jardins	Sao Paulo	SP	4043323
8	5840119709	Gabriel Araujo	R. Manuel de Oliveira	(null)	Santo Amaro	Sao Paulo	SP	80010221

10) Ou aplicar uma consulta, em vez de ser sobre uma tabela, sobre outra consulta. Por exemplo:

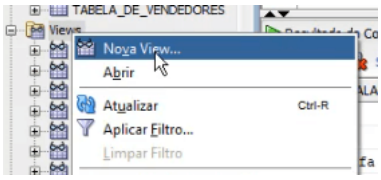
```
SELECT X.EMBALAGEM, X.PRECO_MAXIMO FROM (
    SELECT EMBALAGEM, MAX(PRECO_DE_LISTA) AS PRECO_MAXIMO FROM tabela_de_produtos
    GROUP BY EMBALAGEM
) X
WHERE X.PRECO_MAXIMO >= 10;
```



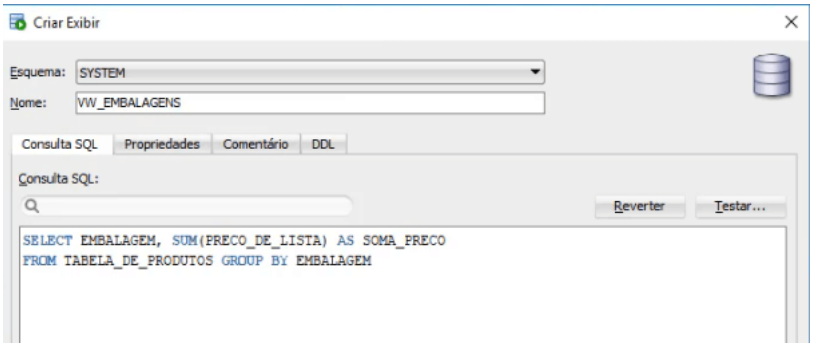
Query Result | SQL | All Rows Fetched: 2 in 0.003 secon

	EMBALAGEM	SOMA_PRECO
1	Lata	19.9935
2	Garrafa	76.9335

11) Você pode transformar uma consulta em uma visão (*view*), que depois pode ser usada em outras consultas, como uma tabela. Crie a visão. Para isso expanda, na árvore do canto esquerdo, onde temos a aba Views e escolha Nova View:

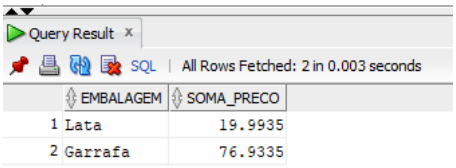


12) Digite as informações conforme a imagem abaixo:



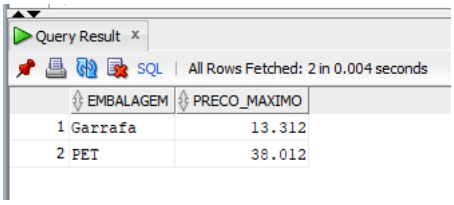
13) Você pode manipular a visão como uma tabela. Por exemplo:

```
SELECT * FROM VW_EMBALAGENS WHERE SOMA_PRECO <= 80;
```



14) Logo, a consulta:

```
SELECT X.EMBALAGEM, X.PRECO_MAXIMO FROM (
    SELECT EMBALAGEM, MAX(PRECO_DE_LISTA) AS PRECO_MAXIMO
    FROM TABELA_DE_PRODUTOS
    GROUP BY EMBALAGEM
) X
WHERE X.PRECO_MAXIMO >= 10;
```



Pode ser substituída por:

```
SELECT X.EMBALAGEM, X.PRECO_MAXIMO FROM
VW_EMBALAGENS X WHERE X.PRECO_MAXIMO >= 10;
```