

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

- 1) Aqui veremos como conectar as consultas de tabelas diferentes. Chamamos esta união de JOIN.
- 2) Veja o conteúdo de duas tabelas digitando os comandos abaixo:

```
SELECT * FROM tabela_de_vendedores;
```

	MATRICULA	NOME	COUNT(*)
▶	00235	Márcio Almeida Silva	29389
	00236	Cláudia Morais	29375
	00237	Roberta Martins	29113

```
SELECT * FROM notas_fiscais;
```

	CPF	MATRICULA	DATA_VENDA	NUMERO	IMPOSTO
▶	7771579779	00235	2015-01-01	100	0.1
	50534475787	00237	2015-01-01	101	0.12
	8502682733	00236	2015-01-01	102	0.12
	5840119709	00235	2015-01-01	103	0.12
	1471156710	00235	2015-01-01	104	0.12
	94387575700	00236	2015-01-01	105	0.1
	3623344710	00237	2015-01-01	106	0.1
	5576228758	00236	2015-01-01	107	0.1
	19290992743	00237	2015-01-01	108	0.12
	94387575700	00236	2015-01-01	109	0.12
	5840119709	00235	2015-01-01	110	0.12
	492472718	00237	2015-01-01	111	0.1
	5840119709	00236	2015-01-01	112	0.12
	50534475787	00235	2015-01-01	113	0.1
	94387575700	00235	2015-01-01	114	0.1
	94387575700	00236	2015-01-01	115	0.12
	50534475787	00235	2015-01-01	116	0.1

- 3) Podemos conectar essas duas tabelas pelo campo em comum (MATRICULA). Digite:

```
SELECT * FROM tabela_de_vendedores A
INNER JOIN notas_fiscais B
ON A.MATRICULA = B.MATRICULA;
```

	MATRICULA	NOME	PERCENTUAL_COMISSAO	DATA_ADMISSAO	DE_FERIAS	BAIRRO	CPF	MATRICULA	DATA_VENDA	NUMERO	IMPOSTO
▶	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	7771579779	00235	2015-01-01	100	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01	103	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01	104	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5840119709	00235	2015-01-01	110	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	50534475787	00235	2015-01-01	113	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	94387575700	00235	2015-01-01	114	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	50534475787	00235	2015-01-01	116	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	2600586709	00235	2015-01-01	118	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	3623344710	00235	2015-01-01	120	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	3623344710	00235	2015-01-01	121	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	94387575700	00235	2015-01-01	124	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5576228758	00235	2015-01-01	133	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	1471156710	00235	2015-01-01	134	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5648641702	00235	2015-01-01	135	0.12
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	5648641702	00235	2015-01-01	138	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	19290992743	00235	2015-01-01	145	0.1
	00235	Márcio Almeida Silva	0.08	2014-08-15	0	Tijuca	94387575700	00235	2015-01-01	149	0.12

4) Podemos aplicar agrupamentos ao resultado da consulta que conecta uma ou mais tabelas:

```
SELECT A.MATRICULA, A.NOME, COUNT(*) FROM  
tabela_de_vendedores A  
INNER JOIN notas_fiscais B  
ON A.MATRICULA = B.MATRICULA  
GROUP BY A.MATRICULA, A.NOME;
```

	MATRICULA	NOME	COUNT(*)
▶	00235	Márcio Almeida Silva	29389
	00236	Cláudia Morais	29375
	00237	Roberta Martins	29113

5) Nem sempre todas as linhas podem ser conectadas. Existem outros tipos de JOINS que nos permite identificar quem não pode ser conectado. Veja a consulta abaixo:

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

Result Grid	
	COUNT(*)
▶	15

Ela mostra que temos 15 clientes.

6) Vamos fazer um JOIN com a tabela de notas fiscais e ver quantos clientes possuem notas emitidas. Digite:

CPF	NOME	CPF
▶ 1471156710	Érica Carvalho	1471156710
19290992743	Fernando Cavalcante	19290992743
2600586709	César Teixeira	2600586709
3623344710	Marcos Nogueira	3623344710
492472718	Eduardo Jorge	492472718
50534475787	Abel Silva	50534475787
5576228758	Petra Oliveira	5576228758
5648641702	Paulo César Mattos	5648641702
5840119709	Gabriel Araujo	5840119709
7771579779	Marcelo Mattos	7771579779
8502682733	Valdeci da Silva	8502682733
8719655770	Carlos Eduardo	8719655770
9283760794	Edson Meilletes	9283760794
94387575700	Walber Lontra	94387575700

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

7) Podemos usar o LEFT JOIN. Digite:

```
SELECT DISTINCT A.CPF, A.NOME, B.CPF FROM tabela_de_clientes A  
LEFT JOIN notas_fiscais B ON A.CPF = B.CPF
```

CPF	NOME	CPF
1471156710	Érica Carvalho	1471156710
19290992743	Fernando Cavalcante	19290992743
2600586709	César Teixeira	2600586709
3623344710	Marcos Nogueira	3623344710
492472718	Eduardo Jorge	492472718
50534475787	Abel Silva	50534475787
5576228758	Petra Oliveira	5576228758
5648641702	Paulo César Mattos	5648641702
5840119709	Gabriel Araújo	5840119709
7771579779	Marcelo Mattos	7771579779
8502682733	Valdeci da Silva	8502682733
8719655770	Carlos Eduardo	8719655770
9283760794	Edson Meilhes	9283760794
94387575700	Walber Lontra	94387575700
95939180787	Fábio Carvalho	NULL

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

8) A seleção correta seria:

```
SELECT DISTINCT A.CPF, A.NOME, B.CPF FROM tabela_de_clientes A
LEFT JOIN notas_fiscais B ON A.CPF = B.CPF
WHERE B.CPF IS NULL;
```

CPF	NOME	CPF
95939180787	Fábio Carvalho	NULL

9) Podemos juntar duas ou mais consultas, Desde que os campos selecionados sejam os mesmos. Digite:

```
SELECT DISTINCT BAIRRO FROM tabela_de_clientes
UNION
SELECT DISTINCT BAIRRO FROM tabela_de_vendedores;
```

BAIRRO
Jardins
Água Santa
Tijuca
Inhauma
Humaitá
Lapa
Santo Amaro
Brás
Cidade Nova
Piedade
Barra da Tij...
Copacabana

10) O UNION ALL não faz a seleção com um DISTINCT. As linhas se repetem se existirem em ambas as tabelas. Digite:

BAIRRO
Jardins
Água Santa
Tijuca
Inhauma
Humaitá
Lapa
Santo Amaro
Brás
Cidade Nova
Piedade
Barra da Tij...
Tijuca
Jardins
Copacabana
Santo Amaro

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

11) Podemos simular o FULL JOIN, que não é suportado pelo MYSQL, usando o LEFT JOIN e RIGHT JOIN com UNION. Digite:

```
SELECT tabela_de_vendedores.BAIRRO,
tabela_de_vendedores.NOME, DE_FERIAS,
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
UNION
SELECT tabela_de_vendedores.BAIRRO,
tabela_de_vendedores.NOME, DE_FERIAS,
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;
```

BAIRRO	NOME	DE_FERIAS	BAIRRO	NOME
Jardins	Cláudia Moraes	1	Jardins	Érica Carvalho
Tijuca	Márcio Almeida Silva	0	Tijuca	César Teixeira
Tijuca	Márcio Almeida Silva	0	Tijuca	Eduardo Jorge
Tijuca	Márcio Almeida Silva	0	Tijuca	Paulo César Mattos
Santo Amaro	Pericles Alves	0	Santo Amaro	Gabriel Araujo
Jardins	Cláudia Moraes	1	Jardins	Valdeci da Silva
Jardins	Cláudia Moraes	1	Jardins	Carlos Eduardo
Copacabana	Roberta Martins	1	NULL	NULL
NULL	NULL	NULL	Água Santa	Fernando Cavalcante
NULL	NULL	NULL	Inhauma	Marcos Nogueira
NULL	NULL	NULL	Humaitá	Abel Silva
NULL	NULL	NULL	Lapa	Petra Oliveira
NULL	NULL	NULL	Brás	Marcelo Mattos
NULL	NULL	NULL	Cidade Nova	Edson Meilletes
NULL	NULL	NULL	Piedade	Walber Lontra
NULL	NULL	NULL	Barra da Tj...	Fábio Carvalho

12) As sub-consultas permitem que possa ser feita seleções usando como critérios outras seleções. Digite:

```
SELECT * FROM tabela_de_clientes WHERE BAIRRO
IN (SELECT DISTINCT BAIRRO FROM tabela_de_vendedores);
```

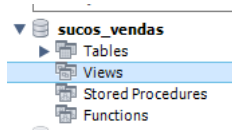
CPF	NOME	ENDERECO_1	ENDERECO_2	BAIRRO	CIDADE	ESTADO	CEP	DATA_DE_NASCIMENTO	IDADE	SEXO	LIMITE_DE_CREDITO	VOLUME_DE_COMPRA	PRIMEIR
1471156710	Érica Carvalho	R. Iriquitã		Jardins	São Paulo	SP	80012212	1990-09-01	27	F	170000	24500	0
2600586709	César Teixeira	Rua Conde de Bonfim		Tijuca	Rio de Janeiro	RJ	22020001	2000-03-12	18	M	120000	22000	0
492472718	Eduardo Jorge	R. Volta Grande		Tijuca	Rio de Janeiro	RJ	22012002	1994-07-19	23	M	75000	9500	1
5648641702	Paulo César Mattos	Rua Hélio Beltrão		Tijuca	Rio de Janeiro	RJ	21002020	1991-08-30	26	M	120000	22000	0
5840119709	Gabriel Araujo	R. Manuel de Oliveira		Santo Amaro	São Paulo	SP	80010221	1985-03-16	32	M	140000	21000	1
8502682733	Valdeci da Silva	R. Srg. Edson de Oliveira		Jardins	São Paulo	SP	82122020	1995-10-07	22	M	110000	19000	0
8719655770	Carlos Eduardo	Av. Gen. Guedes da Fontoura		Jardins	São Paulo	SP	81192002	1983-12-20	34	M	200000	24000	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

13) Podemos aplicar uma consulta, em vez de sobre uma tabela, sobre outra consulta. Digite:

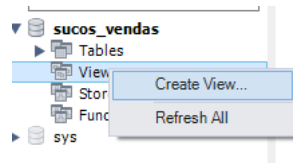
```
SELECT X.EMBALAGEM, X.PRECO_MAXIMO FROM
(SELECT EMBALAGEM, MAX(PRECO_DE_LISTA) AS PRECO_MAXIMO FROM tabela_de_produtos
GROUP BY EMBALAGEM) X WHERE X.PRECO_MAXIMO >= 10;
```

EMBALAGEM	PRECO_MAXIMO
Garrafa	13.312
PET	38.012

14) Podemos transformar uma consulta numa 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 o nome do banco, e vá em Views.



15) Botão da direita do mouse sobre Views e crie uma nova visão.



16) Digite o seguinte comando:

```
CREATE VIEW `VW_MAIORES_EMBALAGENS` AS
SELECT EMBALAGEM, MAX(PRECO_DE_LISTA) AS PRECO_MAXIMO FROM tabela_de_produtos
GROUP BY EMBALAGEM
```

17) Clique em Apply e siga os passos até a criação da visão.

18) Podemos manipular a visão como uma tabela. Digite:

```
SELECT * FROM VW_MAIORES_EMBALAGENS;
```

	EMBALAGEM	PRECO_MAXIMO
▶	Garrafa	13.312
	PET	38.012
	Lata	4.56

19) 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;
```

	EMBALAGEM	PRECO_MAXIMO
▶	Garrafa	13.312
	PET	38.012

Pode ser substituída por:

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

	EMBALAGEM	PRECO_MAXIMO
▶	Garrafa	13.312
	PET	38.012