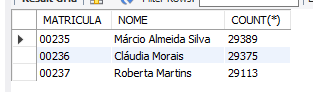
Chegou a hora de você seguir todos os passos realizados por mim durante esta aula. Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

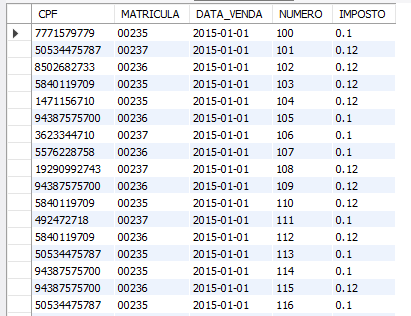
1) Aqui veremos como conectar as consultas de tabelas diferentes. Chamamos esta união de JOIN.

2) Veja o conteúdo de duas tabelas digitando os comandos abaixo:

SELECT \* FROM tabela\_de\_vendedores;COPIAR CÓDIGO



SELECT \* FROM notas\_fiscais;COPIAR CÓDIGO



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;COPIAR CÓDIGO



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

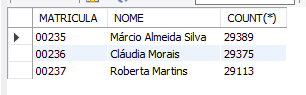
SELECT A.MATRICULA, A.NOME, COUNT(\*) FROM

tabela\_de\_vendedores A

INNER JOIN notas\_fiscais B

ON A.MATRICULA = B.MATRICULA

GROUP BY A.MATRICULA, A.NOME;COPIAR CÓDIGO



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;COPIAR CÓDIGO

5.png

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:

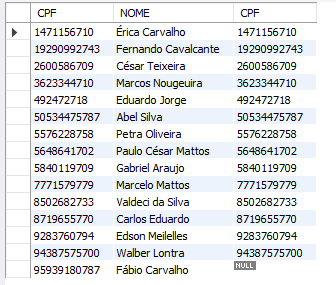


Se você contar verá que, na consulta acima, temos 14 linhas. Existe um cliente que está no cadastro mas não teve nota fiscal emitida.

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.CPFCOPIAR CÓDIGO



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;COPIAR CÓDIGO

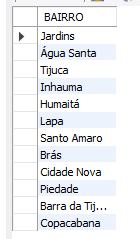
8.png

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;COPIAR CÓDIGO

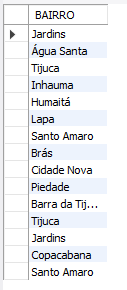


10) O UNION ALL não faz a seleção com um DISTINCT. As linhas se repetem se existirem em ambas as tabelas. Digite:

SELECT DISTINCT BAIRRO FROM tabela\_de\_clientes

UNION ALL

SELECT DISTINCT BAIRRO FROM tabela\_de\_vendedores;COPIAR CÓDIGO



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

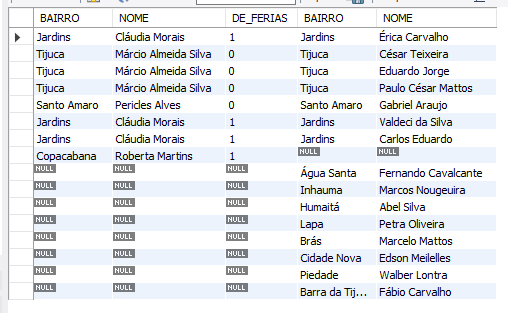
SELECT tabela\_de\_vendedores.BAIRRO,

tabela\_de\_vendedores.NOME, DE\_FERIAS,

tabela\_de\_clientes.BAIRRO,

tabela\_de\_clientes.NOME FROM tabela\_de\_vendedores RIGHT JOIN tabela\_de\_clientes

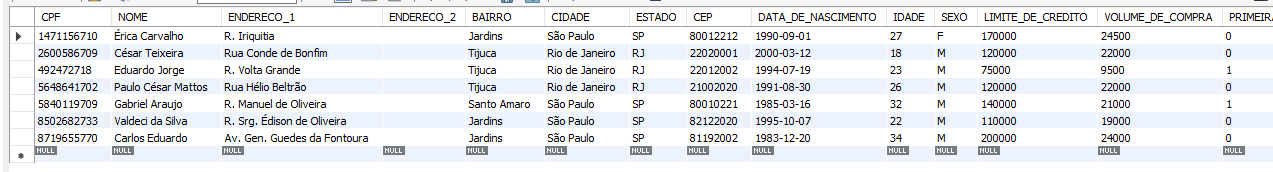
ON tabela\_de\_vendedores.BAIRRO = tabela\_de\_clientes.BAIRRO;COPIAR CÓDIGO



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);COPIAR CÓDIGO

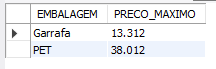


13) Podemos aplicar uma consulta, em vez de sobre uma tabela, sobre outra consulta. Digite:

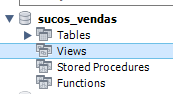
SELECT X.EMBALAGEM, X.PRECO\_MAXIMO FROM

(SELECT EMBALAGEM, MAX(PRECO\_DE\_LISTA) AS PRECO\_MAXIMO FROM tabela\_de\_produtos

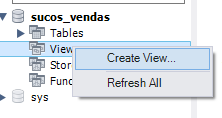
GROUP BY EMBALAGEM) X WHERE X.PRECO\_MAXIMO >= 10;COPIAR CÓDIGO



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

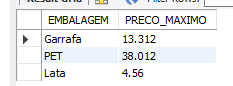
SELECT EMBALAGEM, MAX(PRECO\_DE\_LISTA) AS PRECO\_MAXIMO FROM tabela\_de\_produtos

GROUP BY EMBALAGEMCOPIAR CÓDIGO

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;COPIAR CÓDIGO

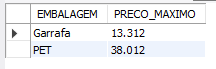


19) Logo a consulta:

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;COPIAR CÓDIGO



Pode ser substituída por:

SELECT X.EMBALAGEM, X.PRECO\_MAXIMO FROM

VW\_MAIORES\_EMBALAGENS X WHERE X.PRECO\_MAXIMO >= 10;COPIAR CÓDIGO

