

Para os exercícios abaixo, utilize a base de dados de venda fornecida pela professora (está no Moodle). Não se esqueça de postar as respostas no Moodle.

```
cliente (codigo, nome, email, telefone)
funcionario (codigo, nome, email, dtaNasc, salario)
venda (numero, data, hora, codclie#, codFun#, tipo)
    codClie REFERENCES cliente (codigo)
    codFun REFERENCES funcionario (codigo)
produto (codigo, nome, preco, qtdEstoque)
produtoVendido (numero#, codProd#, qtd, valor)
    numero REFERENCES venda (numero),
    codProd REFERENCES produto (codigo)
```

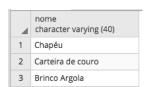
Os exercícios englobam consultas usando todos as possíveis construções vistas em aula: aninhamentos com IN, ANY ou ALL, operadores de conjunto UNION, INTERSECT ou EXCEPT, uso do EXISTS. No entanto, sempre que possível, procure usar o EXISTS para exercitar a sintaxe e o raciocínio em cima do operador.

Antes de executar a consultar, dar update na tabela:

UPDATE produto set preco = 160.000 where codigo = 5; $\Box\Box$

O comando UPDATE é necessário, porque sem ele uma certa consulta errada retornaria o mesmo resultado da consulta correta.

1. Nomes dos produtos que não foram vendidos no período de: início de 2004 até final de 2007. Usando NOT EXISTS.



SELECT p.nome
FROM produto p
WHERE NOT EXISTS (SELECT *

FROM produtovendido pv JOIN venda v on pv.numero = v.numero WHERE p.codigo = pv.codprod AND data between '01-01-2004' AND '12-31-2007')

2. Data e hora das vendas, e nomes dos clientes. As vendas efetuadas para clientes sem cadastro também devem ser listadas.



SELECT data, hora, nome FROM venda LEFT OUTER JOIN cliente on codclie=código

Retornam 30 linhas (não consegui printscreen porque são muitas linhas – não cabem em uma tela)

3. Nomes dos produtos cujo preço de venda seja inferior ao seu preço de custo.



SELECT distinct p.nome --, p.preco, pv.valor FROM produto p JOIN produtovendido pv ON p.codigo = pv.codprod WHERE p.preco > pv.valor

4. Retornar o nome do funcionário que também já foi cliente. Neste caso, uma mesma pessoa é identificada pelo nome + e-mail, ou seja, cliente e funcionário que têm o mesmo nome e o mesmo e-mail são consideradas a mesma pessoa.



a. Usando EXISTS

SELECT nome, email from funcionario f WHERE exists (SELECT *

FROM cliente c WHERE f.nome =c.nome and f.email=c.email)

b. Usando JOIN

SELECT nome, email from funcionario f
WHERE exists (SELECT *



FROM cliente c WHERE f.nome =c.nome and f.email=c.email)

Forma mais fácil de resolver é usando INTERSECT:

SELECT nome, email from funcionario INTERSECT SELECT nome, email from cliente

5. Data de venda, nomes dos produtos e total vendido no período de: início de 2003 até final de 2004, desde que este total seja superior a 100. Ordene por data de venda.

4	data date	nome character varying (40)	sum numeric
1	2004	Jaqueta	264.00

SELECT v.data, p.nome, sum(pv.qtd*pv.valor)

FROM produto p JOIN produtovendido pv ON p.codigo = pv.codprod

JOIN venda v ON pv.numero = v.numero

WHERE data between '01-01-2003' AND '12-31-2004'

GROUP BY v.data, p.nome

HAVING sum(pv.qtd*pv.valor) > 100

6. Data da venda, nome do funcionário que efetuou a venda e total vendido para o cliente 'Monira Rosa'. Ordene por data de venda.

4	data date	nome character varying (40)	sum numeric
1	2001-07-01	Leinaura	35.77
2	2006-07-01	Berlindo	129.00

SELECT v.data, f.nome, sum(pv.qtd*pv.valor)

FROM funcionario f JOIN venda v ON v.codfun = f.codigo

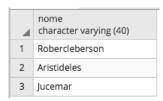
JOIN cliente c ON c.codigo = v.codclie

JOIN produtovendido pv ON pv.numero = v.numero

WHERE c.nome = 'Monira Rosa' GROUP BY v.data, f.nome

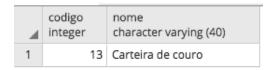


 Selecionar o nome do funcionário sujo salário é maior do que o funcionário mais velho da empresa.



SELECT f.nome
FROM funcionario f
WHERE f.salario > (SELECT salario
FROM funcionario
WHERE dtaNasc = (SELECT min(dtaNasc)
FROM funcionario))

 Nome do produto de maior preço de custo, desde que n\u00e3o tenha sido vendido ainda. Usando NOT EXISTS



```
SELECT p.codigo, NOME□

FROM PRODUTO p □

WHERE p.preco = (SELECT max(a.preco)□

FROM□(SELECT p1.codigo, p1.preco□

FROM produto p1□

WHERE NOT EXISTS (SELECT pv.codprod□

FROM produtovendido pv□

WHERE pv.codprod = p1.codigo) □) a□)
```

2. A consulta abaixo retorna uma dupla do banco: "Leinaura"O que quer dizer este resultado? Explique:

```
SELECT f.nome

FROM funcionario f

WHERE NOT EXISTS (SELECT *

FROM cliente c

WHERE NOT EXISTS (SELECT *

FROM venda v

WHERE v.codclie =c.codigo AND v.codfun = f.codigo))
```



Universidade Federal de Santa Catarina – UFSC Departamento de Informática e Estatística – INE/CTC

Prof. Carina Friedrich Dorneles dorneles@inf.ufsc.br

Retorna nome dos funcionários para os quais não existe nenhum cliente para o qual o funcionário não tenha realizado uma venda, ou seja, retorna o nome dos funcionários que venderam para todos os clientes.