

```
/* 1 - ALTERE A TABELA DE DÉBITOS PARA CORRIGIR O CAMPO  
SITUACAO_DEBITO. SE O DÉBITO  
* ESTIVER VENCIDO E NÃO FOR 'NEGOCIADO' NEM 'PAGO', INFORME  
'ATRASADO' */
```

```
UPDATE DEBITOS
```

```
SET SITUACAO_DEBITO = 'Atrasado'
```

```
WHERE DATAVENCIMENTO < CURRENT_DATE
```

```
AND SITUACAO_DEBITO NOT IN ('Negociado', 'Pago')
```

```
/* 2 - ALTERE A TABELA DE DÉBITOS PARA CORRIGIR O CAMPO  
SITUACAO_DEBITO. SE O DÉBITO  
* NÃO ESTIVER VENCIDO E NÃO FOR 'PAGO', INFORME 'PENDENTE' */
```

```
UPDATE DEBITOS
```

```
SET SITUACAO_DEBITO = 'Pendente'
```

```
WHERE DATAVENCIMENTO > CURRENT_DATE
```

```
AND SITUACAO_DEBITO NOT IN ('Pago')
```

```
/* 3 - ALTERE A TABELA DE DÉBITOS PARA CORRIGIR O CAMPO DATA DE  
VENCIMENTO. SE O DÉBITO  
* ESTIVER COM A SITUAÇÃO 'NEGOCIADO', ALTERE O ANO DA DATA DE  
VENCIMENTO PARA ANO - 5 ANOS... */
```

```
UPDATE DEBITOS
```

```
SET DATAVENCIMENTO = DATEADD(-5 YEAR TO DATAVENCIMENTO)
```

```
WHERE SITUACAO_DEBITO = 'Negociado'
```

```
/* 4 - SELECIONE O NOME, E-MAIL E TELEFONE DE TODAS AS PESSOAS */
```

```
SELECT PESSOA.NOME, PESSOA.EMAIL, PESSOA.TELEFONE
```

```
FROM PESSOA
```

```
/* 5 - LISTE OS PRODUTOS COM PREÇO UNITÁRIO ACIMA DE R$ 100 */
```

```
SELECT PRODUTO.NOME, PRODUTO.PRECO_UNITARIO
```

```
FROM PRODUTO WHERE PRECO_UNITARIO > 100
```

```
/* 6 - RETORNE TODAS AS VENDAS REALIZADAS EM UMA DETERMINADA DATA */
```

```
SELECT * FROM VENDA
```

```
WHERE DATA_VENDA BETWEEN '2022-12-01' AND '2023-12-01'
```

```
/* 7 - MOSTRE O NOME, A DATA DE NASCIMENTO E A IDADE DOS  
FORNECEDORES QUE NÃO TÊM PRODUTOS EM ESTOQUE */
```

```
SELECT PESSOA.NOME, PESSOA.DATA_NASCIMENTO, ((CURRENT_DATE -  
PESSOA.DATA_NASCIMENTO)/365)
```

```
FROM FORNECEDOR
```

```
LEFT JOIN PESSOA ON PESSOA.ID_PESSOA = FORNECEDOR.ID_PESSOA
WHERE FORNECEDOR.ID_FORNECEDOR IN (SELECT ID_FORNECEDOR FROM PRODUTO
WHERE QUANTIDADE_ESTOQUE IS NULL)
```

```
/* 8 - LISTE OS CLIENTES QUE JÁ REALIZARAM COMPRAS COM DÉBITOS
VENCIDOS */
```

```
SELECT PESSOA.NOME FROM DEBITOS
INNER JOIN CLIENTE ON CLIENTE.ID_CLIENTE = DEBITOS.ID_CLIENTE
INNER JOIN PESSOA ON PESSOA.ID_PESSOA = CLIENTE.ID_PESSOA
WHERE DEBITOS.SITUACAO_DEBITO = 'Atrasado'
```

```
/* 9 - SELECIONE OS CLIENTES QUE POSSUEM ITENS NO CARRINHO DE
COMPRAS E POSSUEM DÉBITOS VENCIDOS - NÃO É NECESSÁRIO FAZER */
```

```
/* 10 - SELECIONE OS 10 MAIORES DEVEDORES TRAZENDO CÓDIGO, NOME,
IDADE E A SOMA TOTAL DOS DÉBITOS
* VENCIDOS E A SOMA TOTAL DOS DÉBITOS PAGOS */
```

```
SELECT FIRST 10 CLIENTE.ID_CLIENTE, PESSOA.NOME, EXTRACT(YEAR FROM
CURRENT_DATE) - EXTRACT(YEAR FROM PESSOA.DATA_NASCIMENTO) AS IDADE
, COALESCE((SELECT SUM(DEBITOS.VALOR_TOTAL) FROM DEBITOS
WHERE DEBITOS.ID_CLIENTE = CLIENTE.ID_CLIENTE
AND DEBITOS.DATAVENCIMENTO < CURRENT_DATE
AND DEBITOS.SITUACAO_DEBITO IN ('Atrasado', 'Pendente')),0) AS
DEBITO_PENDENTE
, (SELECT SUM(DEBITOS.VALOR_TOTAL) FROM DEBITOS
WHERE DEBITOS.ID_CLIENTE = CLIENTE.ID_CLIENTE
AND SITUACAO_DEBITO = 'Pago') AS DEBITO_PAGO FROM PESSOA
INNER JOIN CLIENTE ON CLIENTE.ID_PESSOA = PESSOA.ID_PESSOA
ORDER BY 4 DESC
```

```
/* 11 - ATUALIZE O PREÇO UNITÁRIO DE TODOS OS PRODUTOS COM ESTOQUE
INFERIOR A 10 UNIDADES PARA MAIS 10% */
```

```
UPDATE PRODUTO SET PRECO_UNITARIO = PRECO_UNITARIO * 1.1
WHERE QUANTIDADE_ESTOQUE < 10
```

```
/* 12 - MODIFIQUE O STATUS DE ENTREGA PARA 'CONCLUÍDO' EM TODAS AS
ENTREGAS REALIZADAS ANTES DE '2024-01-01' - NÃO É NECESSÁRIO FAZER
*/
```

```
/* 13 - ALTERE TODOS OS DÉBITOS QUE ESTÃO COM SITUAÇÃO = 'ATRASADO'
PARA 'PAGOS' SE O MESMO TER O
```

/* REGISTRO NA TABELA DE PAGAMENTO */

```
UPDATE DEBITOS SET SITUACAO_DEBITO = 'Pagos'  
WHERE SITUACAO_DEBITO = 'Atrasado' AND ID_DEBITO IN (SELECT  
ID_DEBITO FROM PAGAMENTO)
```

/* 14 - SELECIONE AS VENDAS ENTRE O MÊS DE DEZEMBRO DE 2023 ATÉ JANEIRO DE 2024 */

```
SELECT * FROM VENDA  
WHERE DATA_VENDA BETWEEN '2023-12-01' AND '2024-01-01'
```

/* 15 - LISTE OS PRODUTOS COM ESTOQUE ENTRE 5 E 20 UNIDADES */

```
SELECT PRODUTO.NOME, PRODUTO.QUANTIDADE_ESTOQUE FROM PRODUTO  
WHERE QUANTIDADE_ESTOQUE BETWEEN 5 AND 20
```

/* 16 - RETORNE TODAS AS PESSOAS COM NOMES QUE COMEÇAM COM 'A' */

```
SELECT PESSOA.NOME FROM PESSOA  
WHERE NOME LIKE 'A%'
```

/* 17 - SELECIONE OS PRODUTOS QUE NÃO FORAM VENDIDOS */

```
SELECT PRODUTO.ID_PRODUTO FROM PRODUTO  
LEFT JOIN ITENSVENDA ON PRODUTO.ID_PRODUTO = ITENSVENDA.ID_PRODUTO  
WHERE ITENSVENDA.ID_PRODUTO IS NULL
```

/* 18 - LISTE OS PRODUTOS VENDIDOS PARA OS CLIENTES 'CARLA SOUZA', 'LARISSA FERREIRA' */

```
SELECT ID_PESSOA FROM VENDA  
WHERE ID_PESSOA = 3 OR ID_PESSOA = 11
```

/* 19 - MOSTRE AS PESSOAS QUE SÃO CLIENTES OU FORNECEDORES */

```
SELECT PESSOA.NOME FROM PESSOA  
JOIN FORNECEDOR ON PESSOA.ID_PESSOA = FORNECEDOR.ID_PESSOA
```

/* 20 - RETORNE TODAS AS INFORMAÇÕES DA VENDA COM DETALHES DO CLIENTE E FORMA DE PAGAMENTO */

```
SELECT * FROM VENDA  
LEFT JOIN PESSOA ON PESSOA.ID_PESSOA = VENDA.ID_PESSOA  
LEFT JOIN PAGAMENTO ON PAGAMENTO.ID_FORMA_PAGAMENTO =  
VENDA.ID_FORMA_PAGAMENTO
```

/* 21 - MOSTRE TODAS AS COMPRAS DE UM DETERMINADO PRODUTO ('PLACA DE CAPTURA'), INCLUINDO DETALHES DO PRODUTO */

```
SELECT QUANTIDADE FROM ITENSVENDA
```

```
LEFT JOIN VENDA ON VENDA.ID_VENDA = ITENSVENDA.ID_VENDA
WHERE VENDA.ID_VENDA = 17
ARRUMARRR
```

```
SELECT * FROM VENDA WHERE ID_VENDA = 17
JOIN QUANTIDADE ON ITENSVENDA.ID_VENDA = VENDA.ID_VENDA
ARRUMARRR
```

```
/* 22 - COMBINE OS NOMES DOS FORNECEDORES COM OS NOMES DOS CLIENTES
E SUAS RESPECTIVAS IDADES
```

```
* QUE IRÃO FAZER NO ANO DE 2025 */
```

```
SELECT FORNECEDOR.ID_PESSOA AS ID_FORNECEDOR, CLIENTE.ID_PESSOA AS
ID_CLIENTE, (2025 - EXTRACT (YEAR FROM PESSOA.DATA_NASCIMENTO)) AS
IDADE_2025, PESSOA.NOME FROM PESSOA
LEFT JOIN FORNECEDOR ON FORNECEDOR.ID_PESSOA = PESSOA.ID_PESSOA
LEFT JOIN CLIENTE ON CLIENTE.ID_PESSOA = PESSOA.ID_PESSOA
```

```
/* 23 - RETORNE TODOS OS LOGRADOUROS E BAIRROS, INDICANDO SE
PERTENCEM OU NÃO A UMA PESSOA */
```

```
SELECT LOGRADOURO.NOME AS LOGRADOURO, BAIRRO.NOME AS BAIRRO,
ENDereco.ID_PESSOA FROM ENDereco
LEFT JOIN LOGRADOURO ON LOGRADOURO.ID_LOGRADOURO =
ENDereco.ID_LOGRADOURO
LEFT JOIN BAIRRO ON BAIRRO.ID_BAIRRO = ENDereco.ID_BAIRRO
```

```
/* 24 - SELECIONE OS PRODUTOS QUE TÊM PREÇO UNITÁRIO SUPERIOR À
MÉDIA */
```

```
SELECT PRODUTO.PRECO_UNITARIO FROM PRODUTO
GROUP BY PRODUTO.PRECO_UNITARIO
HAVING PRODUTO.PRECO_UNITARIO > AVG(PRODUTO.PRECO_UNITARIO)
```

```
/* 25 - LISTE OS CLIENTES QUE FIZERAM COMPRAS EM CIDADES COM MAIS DE
1 MILHÃO DE HABITANTES */
```

```
SELECT CLIENTE.ID_PESSOA, HISTORICO_COMPRAS_CLIENTE.ID_VENDA,
PESSOA.NOME, ENDereco.ID_MUNICIPIO, MUNICIPIO.POPULACAO FROM
MUNICIPIO
LEFT JOIN ENDereco ON ENDereco.ID_MUNICIPIO = MUNICIPIO.ID_MUNICIPIO
LEFT JOIN CLIENTE ON CLIENTE.ID_PESSOA = ENDereco.ID_PESSOA
LEFT JOIN PESSOA ON PESSOA.ID_PESSOA = CLIENTE.ID_PESSOA
INNER JOIN HISTORICO_COMPRAS_CLIENTE ON
HISTORICO_COMPRAS_CLIENTE.ID_CLIENTE = CLIENTE.ID_CLIENTE
WHERE POPULACAO > 1000000
```

```

/* 26 - MOSTRE A QUANTIDADE TOTAL DE PRODUTOS VENDIDOS POR
FORNECEDOR */
SELECT FORNECEDOR.ID_FORNECEDOR, PESSOA.NOME AS NOME_FORNECEDOR,
SUM(ITENSVENDA.QUANTIDADE) AS TOTAL_PRODUTOS_VENDIDOS FROM
FORNECEDOR
INNER JOIN PESSOA ON PESSOA.ID_PESSOA = FORNECEDOR.ID_PESSOA
INNER JOIN PRODUTO ON FORNECEDOR.ID_FORNECEDOR =
PRODUTO.ID_FORNECEDOR
INNER JOIN ITENSVENDA ON PRODUTO.ID_PRODUTO = ITENSVENDA.ID_PRODUTO
GROUP BY FORNECEDOR.ID_FORNECEDOR, PESSOA.NOME

/* 27 - LISTE OS CLIENTES QUE REALIZARAM COMPRAS COM VALOR TOTAL
SUPERIOR A R$ 500 */
SELECT ID_PESSOA FROM VENDA
WHERE VALOR_TOTAL > 500

/* 28 - RETORNE TODAS AS PESSOAS COM SEUS ENDEREÇOS DE
CORRESPONDÊNCIA E RESIDÊNCIA, SE TIVEREM */
SELECT PESSOA.NOME, ENDereco.* FROM PESSOA
JOIN ENDereco ON ENDereco.ID_PESSOA = PESSOA.ID_PESSOA

/* 29 - LISTE OS PRODUTOS E SEUS PREÇOS, JUNTAMENTE COM O NOME DO
FORNECEDOR, MESMO QUE NÃO TENHAM FORNECEDOR */
SELECT PRODUTO.NOME, PRODUTO.PRECO_UNITARIO, PESSOA.NOME FROM
PRODUTO
LEFT JOIN FORNECEDOR ON PRODUTO.ID_FORNECEDOR =
FORNECEDOR.ID_FORNECEDOR
LEFT JOIN PESSOA ON FORNECEDOR.ID_PESSOA = PESSOA.ID_PESSOA
ORDER BY PESSOA.NOME

/* 30 - LISTE OS PRODUTOS EM ORDEM DECRESCENTE DE QUANTIDADE EM
ESTOQUE */
SELECT PRODUTO.NOME, PRODUTO.QUANTIDADE_ESTOQUE FROM PRODUTO
ORDER BY PRODUTO.QUANTIDADE_ESTOQUE DESC

/* 31 - RETORNE AS VENDAS EM ORDEM CRESCENTE DE VALOR TOTAL */
SELECT VALOR_TOTAL FROM VENDA
ORDER BY VENDA.VALOR_TOTAL

/* 32 - LISTE OS PRODUTOS COM QUANTIDADE EM ESTOQUE MENOR QUE A
MÉDIA DE TODOS OS PRODUTOS */

```

```
SELECT PRODUTO.QUANTIDADE_ESTOQUE FROM PRODUTO
GROUP BY PRODUTO.QUANTIDADE_ESTOQUE
HAVING PRODUTO.QUANTIDADE_ESTOQUE < AVG(PRODUTO.QUANTIDADE_ESTOQUE)
```

```
/* 33 - SELECIONE TODAS AS PESSOAS CUJO NÚMERO DE TELEFONE SEJA
IGUAL AO DE OUTRA PESSOA */
```

```
SELECT DISTINCT PESSOA1.* FROM PESSOA PESSOA1
JOIN PESSOA PESSOA2 ON PESSOA1.TELEFONE = PESSOA2.TELEFONE
WHERE PESSOA1.ID_PESSOA <> PESSOA2.ID_PESSOA
```

```
/* 34 - AUMENTE EM 10% O PREÇO UNITÁRIO DE TODOS OS PRODUTOS
FORNECIDOS POR FORNECEDORES DE SÃO PAULO */
```

```
UPDATE PRODUTO SET PRODUTO.PRECO_UNITARIO = PRODUTO.PRECO_UNITARIO *
1.1
```

```
WHERE PRODUTO.ID_FORNECEDOR IN (
SELECT FORNECEDOR.ID_FORNECEDOR
FROM FORNECEDOR
INNER JOIN PESSOA ON FORNECEDOR.ID_PESSOA = PESSOA.ID_PESSOA
INNER JOIN ENDERECO ON PESSOA.ID_PESSOA = ENDERECO.ID_PESSOA
INNER JOIN MUNICIPIO ON ENDERECO.ID_MUNICIPIO =
MUNICIPIO.ID_MUNICIPIO
WHERE MUNICIPIO.NOME = 'São Paulo'
)
```

```
/* 35 - LISTA TODAS AS VENDAS QUE ESTÃO COM DÉBITO 'NEGOCIADO' */
```

```
SELECT ID_VENDA FROM DEBITOS
WHERE SITUACAO_DEBITO = 'Negociado'
```