**Exercícios de SQL para fixação 001**

1. Construa um SQL que retorna o código do produto, descrição do produto [tabela Produto] código da marca e descrição da marca [tabela produto\_marca], código medida venda, descrição medida venda e sigla [tabela medida];

* Trazer todos os produtos, mesmo que não tenha marca ou medida de venda informada;
* Utilizar apelido “alias” nas tabelas e nas colunas;
* Utilizar **join** na junção de tabelas;



select p.codigo,

p.descricao as DESCRICAO\_PRODUTO,

pm.codigo as COD\_MARCA,

pm.descricao as DESCRICAO\_MARCA,

m.codigo as CODIGO\_MEDIDA\_VENDA,

m.descricao as DESCRICAO\_MEDIDA\_VENDA,

m.sigla

from produto p

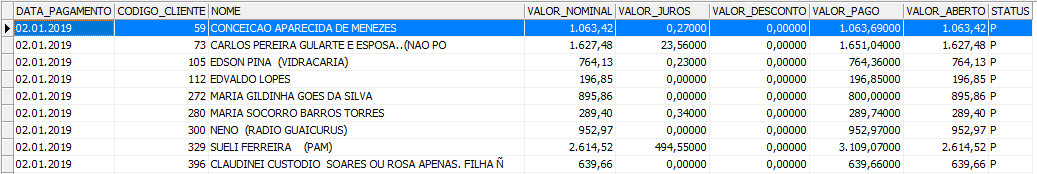
left join produto\_marca pm on pm.codigo = p.cod\_marca

left join medidas m on m.codigo = p.codigo\_medida\_venda

order by p.codigo;

1. Listar e **somar** todos os pagamentos efetuados entre as datas 01/01/2019 até 28/02/2019 e que o valor pago seja igual ou maior a 100,00 . Colunas a serem exibidas: Data pagamento, codigo do cliente [tabela contas\_receber\_pagamento], nome do cliente [tabela cliente], valor\_nominal, valor\_juros, valor\_desconto, valor\_pago, valor\_aberto[tabela contas\_receber\_pagamento], status [tabela contas\_receber].

* A coluna status possui um dos seguintes valores (A, P, N) A = Aberto, P = Pago e N = Negociado. Utilizar a estrutura **CASE**, no corpo do SQL, e fazer retornar a descrição “Aberto ou Pago ou Negociado”.



select

crp.data\_pagamento,

C.codigo,

C.nome,

sum (cr.valor\_nominal) as VALOR\_NOMINAL,

SUM (crp.valor\_juros) AS VALOR\_JUROS,

SUM (CRP.valor\_desconto) AS VALOR\_DESCONTO,

SUM (CRP.valor\_pago) AS VALOR\_PAGO,

SUM ( CRP.valor\_aberto) AS VALOR\_ABERTO,

CASE CR.status

WHEN 'P' THEN 'PAGO'

when 'N' THEN 'NEGOCIADO'

WHEN 'A' THEN 'ABERTO'

else

'SEM INFORMAÇÃO'

END AS STATUS

from contas\_receber\_pagamento crp

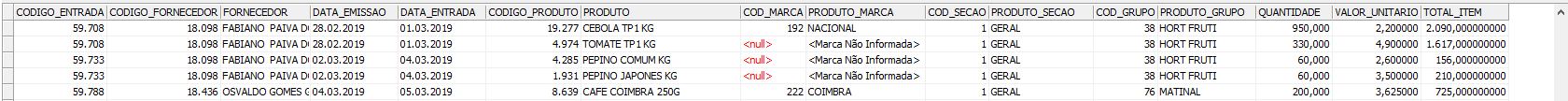
INNER JOIN contas\_receber CR ON CR.codigo = CRP.codigo\_cr

INNER JOIN CLIENTE C ON C.codigo = CR.codigo\_cliente

WHERE CRP.data\_pagamento between '02.01.2019' AND '28.02.2019'

GROUP by CRP.data\_pagamento,C.CODIGO, C.nome, CR.status

1. Listar todas entradas de produtos que ocorreram no período 01/03/2019 até 10/03/2019. Retornar os seguintes campos: codigo\_entrada, codigo\_fornecedor [tab. Entrada], fornecedor “nome”[TAB. fornecedor], data de emissão, data entrada [tab. Entrada], codigo produto, produto “descricao” [tab. prosuto], codigo marca, descrição da marca [tab. Produto marca], codido seção, descrição seção [tab produto\_secao], codigo do grupo descrição do grupo [tab. Produto\_grupo], quantidade, valor unitário e valor total [tab. Entrada\_item];

* Iniciar com o from na tabela entrada\_item e utilizar JOIN para buscar informações das demais tabelas;
* A coluna Total\_Item é calculada pela (quantidade vezes o valor unitário);
* Pesquisar o comando **COALESCE** e utilizar nas colunas que retornam a descrição da marca, seção e grupo. Ou seja, se não houver marca informada retornar '<Marca Não Informada>' sucessivamente para seção e grupo.
* Sempre trazer todos os produtos.

SELECT

E.codigo AS CODIGO\_ENTRADA,

E.codigo\_fornecedor,

F.nome AS NOME\_FORNECEDOR,

E.data\_emissao,

E.data\_entrada,

p.codigo as codigo\_do\_produto,

p.descricao as PRODUTO,

PM.codigo AS COD\_MARCA,

coalesce (PM.descricao, 'Marca Não Informada') AS MARCA,

PS.codigo as COD\_SECAO,

coalesce (PS.descricao,' Seção Não Informada ') AS SECAO,

PG.codigo AS COD\_GRUPO,

coalesce (PG.descricao,'Grupo Não Informado') as GRUPO,

EI.quantidade,

EI.valor\_unitario,

SUM ( EI.quantidade \* EI.valor\_unitario) AS VALOR\_TOTAL

FROM ENTRADA\_ITEM EI

INNER JOIN PRODUTO P ON P.codigo= EI.codigo\_produto

INNER JOIN ENTRADA E ON E.codigo = EI.codigo\_entrada

INNER JOIN FORNECEDOR F ON F.codigo = E.codigo\_fornecedor

left JOIN PRODUTO\_MARCA PM ON PM.codigo = P.cod\_marca

left JOIN produto\_secao PS ON PS.codigo = P.cod\_secao

left JOIN produto\_grupo PG ON PG.CODIGO = P.cod\_grupo

WHERE E.data\_entrada between '01.03.2019' AND '10.03.2019'

GROUP BY

E.codigo,

E.codigo\_fornecedor,

F.nome ,

E.data\_emissao,

E.data\_entrada,

p.codigo ,

p.descricao,

PM.codigo ,

PM.descricao ,

PS.codigo ,

PS.descricao ,

PG.codigo ,

PG.descricao ,

EI.quantidade,

EI.valor\_unitario

ORDER BY P.descricao