

```

1  /*****
2  Aula 6 - 18/abril/19
3  *****/
4  /**** Atividade 4
5  7 - Alterar a tabela Itens de Pedido incluindo uma nova coluna preco_item.
6  Atualize o valor dessa coluna para que os itens tenham preço 0,5% menor que o valor original do produto
7  ( está na tabela produto).
8  Após essas alterações elabore um controle para evitar que o preço do item de um produto -
9  ao ser inserido ou atualizado- seja maior que o preço original do produto. */
10 ALTER TABLE itens_pedido DROP COLUMN num_item ;
11
12 SELECT * FROM itens_pedido ;
13 -- alterando a tabela itens_pedido para construir a procedure abaixo
14 ALTER TABLE itens_pedido ADD preco_item NUMBER(10,2) ;
15 ALTER TRIGGER valida_estoque DISABLE ;
16 UPDATE itens_pedido i SET i.preco_item = ( SELECT p.preco_venda*0.995 FROM produto p
17 WHERE p.cod_prod = i.cod_prod ) ;
18
19 -- gatilho para validar o preço do item
20 CREATE OR REPLACE TRIGGER valida_preco_item
21 BEFORE INSERT OR UPDATE ON itens_pedido
22 FOR EACH ROW
23 DECLARE
24 vpreco produto.preco_venda%TYPE ;
25 PRAGMA AUTONOMOUS_TRANSACTION ;
26 BEGIN
27 select preco_venda into vpreco
28 from produto
29 where cod_prod = :NEW.cod_prod ;
30 IF :NEW.preco_item > vpreco OR :NEW.preco_item = 0 THEN
31 raise_application_error ( -20032, 'Preço do item não pode ser maior que o sugerido para o produto nem ser de graça!!');
32 END IF ;
33 END;
34
35 -- teste
36 SELECT get_preco( 5010) FROM dual ;
37 INSERT INTO itens_pedido VALUES ( 1905 , 5010, 1, 0, 'SEPARACAO', 108, null );
38
39
40 /****
41 8 - Elaborar um extrato do pedido no seguinte formato (faça de duas maneiras : com cursor FETCH e cursor FOR):
42 Pedido : 1906 Data : 01/MAR/2019 Cliente : José Arimatéia Forma Pagamento : Cartao Credito
43 -----
44 Item Produto Preço Item Qtde Pedida Valor Item Valor Pedido
45 1 Bola Futebol 30,00 3 90,00 90,00
46 2 Meia Futebol 19,00 1 19,00 109,00
47 3 Camisa Seleção 150,00 2 300,00 409,00
48 -----
49 Total Pedido : 409,00 Vendedor : Lúcia Castro ***/
50
51
52 -- usando cursor fetch -- funcionou 10/abril
53 CREATE OR REPLACE PROCEDURE cupompedido (vped IN INTEGER)
54 IS
55 CURSOR cupom IS
56 SELECT i.*, pr.nome_prod
57 FROM itens_pedido i, produto pr
58 where i.cod_prod = pr.cod_prod
59 and i.num_ped = vped
60 ORDER BY pr.nome_prod ;
61
62 pedidolinha cupom%ROWTYPE ;
63 vdata pedido.dt_hora_ped%TYPE ;
64 vcliente cliente.nome_fantasia%TYPE ;
65 vpgto forma_pgto.descr_forma_pgto%TYPE ;
66 vvendedor funcionario.nome_func%TYPE ;
67 vtotallinha pedido.vl_total_ped%TYPE ;
68 separador VARCHAR2(100) := RPAD('-',99,'-');
69 cabecalho VARCHAR2(100) ;
70 vindice SMALLINT := 1 ;
71 BEGIN
72 select c.nome_fantasia, p.dt_hora_ped, fp.descr_forma_pgto, f.nome_func
73 INTO vcliente, vdata, vpgto, vvendedor
74 from cliente c, pedido p, funcionario f, forma_pgto fp
75 where c.cod_cli = p.cod_cli
76 and p.reg_func_vendedor = f.reg_func
77 and p.forma_pgto = fp.cod_forma
78 and p.num_ped = vped ;
79
80 cabecalho := 'Pedido : ' || TO_CHAR (vped) || ' Data : ' ||
81 TO_CHAR(vdata, 'DD-MON-YYYY')
82 || ' Cliente : ' || vcliente || ' Forma Pagto: ' || vpgto;
83 DBMS_OUTPUT.put_line ( cabecalho );
84 DBMS_OUTPUT.put_line ( separador );
85 DBMS_OUTPUT.put_line ( ' Item | Produto |
86 Preço Item | Qtde Pedida | Valor Item | Valor Pedido' );
87 vtotallinha := 0 ;
88 OPEN cupom ;
89 LOOP
90 FETCH cupom INTO pedidolinha ;
91 EXIT WHEN cupom%NOTFOUND ;
92
93 vtotallinha := vtotallinha + pedidolinha.qtde_pedida*pedidolinha.preco_item;
94
95 DBMS_OUTPUT.put_line (
96 TO_CHAR (vindice, '009999') ||
97 ' | ' || RPAD(pedidolinha.nome_prod,29,' ') || ' | ' ||
98 TO_CHAR(pedidolinha.preco_item, '$999999D99') || ' | ' ||
99 TO_CHAR(pedidolinha.qtde_pedida, '9999') || ' | ' ||
100 TO_CHAR(pedidolinha.preco_item*pedidolinha.qtde_pedida, '$999999D99') || ' | ' ||
101 TO_CHAR(vtotallinha, '$999999D99') );
102 vindice := vindice + 1 ;
103 END LOOP ;
104 CLOSE cupom ;
105 DBMS_OUTPUT.put_line ( separador );

```

```

106 DBMS_OUTPUT.put_line ( 'Total Pedido : ' ||
107 TO_CHAR (vtotallinha, '$99999999D999') ||
108 ' Vendedor : ' || vvendedor );
109 END ;
110 -- executando
111 BEGIN
112 cupompedido (1906) ;
113 END ;
114
115 select * from itens_pedido;
116
117 -- usando cursor FOR -- funcionou 10/abril
118 CREATE OR REPLACE PROCEDURE cupompedido_for (vped IN INTEGER)
119 IS
120     CURSOR cupomfor IS
121     SELECT i.*, pr.nome_prod
122     FROM itens_pedido i, produto pr
123     where i.cod_prod = pr.cod_prod
124     and i.num_ped = vped ;
125
126 --pedidolinha cupom%ROWTYPE ;
127 vdata pedido.dt_hora_ped%TYPE ;
128 vcliente cliente.nome_fantasia%TYPE ;
129 vpgto forma_pgto.descr_forma_pgto%TYPE ;
130 vvendedor funcionario.nome_func%TYPE ;
131 vtotallinha pedido.vl_total_ped%TYPE ;
132 cabecalho VARCHAR2(100) ;
133 vindice SMALLINT := 1 ;
134 BEGIN
135     select c.nome_fantasia, p.dt_hora_ped, fp.descr_forma_pgto, f.nome_func
136     INTO vcliente, vdata, vpgto, vvendedor
137     from cliente c, pedido p, funcionario f, forma_pgto fp
138     where c.cod_cli = p.cod_cli
139     and p.reg_func_vendedor = f.reg_func
140     and p.forma_pgto = fp.cod_forma
141     and p.num_ped = vped ;
142
143 cabecalho := 'Pedido : ' || TO_CHAR (vped) || 'Data : ' || TO_CHAR(vdata, 'DD-MON-YYYY') || ' Cliente : ' || vcliente || ' Forma Pagto: ' || vpgto;
144 DBMS_OUTPUT.put_line ( cabecalho ) ;
145 DBMS_OUTPUT.put_line ( ' Item | Produto | Preço Item | Qtde Pedida | Valor Item | Valor Pedido' ) ;
146 vtotallinha := 0 ;
147 FOR j IN cupomfor LOOP
148
149     vtotallinha := vtotallinha + j.qtde_pedida*j.preco_item;
150
151     DBMS_OUTPUT.put_line ( TO_CHAR (vindice, '009999') || ' ' || RPAD(j.nome_prod, 30, ' ') ||
152 TO_CHAR(j.preco_item, '999999D99') || ' ' || TO_CHAR(j.qtde_pedida, '9999') || ' ' ||
153 TO_CHAR(j.preco_item*j.qtde_pedida, '9999999D99') ||
154 TO_CHAR(vtotallinha, '9999999D99') ) ;
155     vindice := vindice + 1 ;
156 END LOOP ;
157 --CLOSE cupom ;
158 DBMS_OUTPUT.put_line ( 'Total Pedido : ' ||
159 TO_CHAR (vtotallinha, '99999999D99') ||
160 ' Vendedor : ' || vvendedor );
161 END ;
162
163 BEGIN
164 cupompedido_for (1906) ;
165 END ;

```