

```

1  -- Programação BD Vendas Produtos Esportivos
2  -- Versao para Oracle 11g
3  /*
4  cliente ( cod_cli(PK), limite_credito, endereco_cli, fone_cli, situacao_cli, tipo_cli, cod_regiao(fk))
5  cliente_pf (cod_cli_pf(PK)(FK), nome_fantasia, cpf_cli, sexo_cli, profissao_cli)
6  cliente_pj (cod_cli_pj(PK)(FK), razao_social_cli, cnpj_cli, ramo_atividade_cli)
7  produto ( cod_prod(PK), nome_prod, descr_prod, categ_esporte, preco_venda, preco_custo, peso, marca, tamanho)
8  funcionario ( reg_func(PK), nome_func, end_func, depto, sexo_func, dt_admissao, cargo, cod_regiao(fk))
9  regiao (cod_regiao(PK), nome_regiao)
10 deposito ( cod_depo(PK), nome_depo, end_depo, cidade_depo, pais_depo, cod_regiao(fk))
11 pedido ( num_ped(PK), dt_hora_ped, tp_atendimento, vl_total_ped, vl_descto, vl_frete,
12 end_entrega, forma_pgto, cod_cli(fk), reg_func_vendedor(fk))
13 itens_pedido (num_ped(FK)(PK), cod_prod(fk)(PK), qtde_item, descto_item)
14 armazenamento ( cod_depo(FK)(PK), cod_prod(FK)(PK), qtde_estoque, end_estoque)*/
15
16
17 /* parametros de configuracao da sessao */
18 ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY HH24:MI:SS';
19 ALTER SESSION SET NLS_LANGUAGE = PORTUGUESE;
20 SELECT SESSIONTIMEZONE, CURRENT_TIMESTAMP FROM DUAL;
21
22 --Aula FATEC 21/fev/2019 -Validar a inserção de um produto como item desde que a quantidade pedida
23 --seja menor ou igual à quantidade em estoque para o depósito que atende à região do cliente, ou seja,
24 --não será possível cadastrar um item no pedido se não houver quantidade suficiente no estoque.
25 CREATE OR REPLACE TRIGGER valida_estoque
26 BEFORE INSERT OR UPDATE ON itens_pedido
27 FOR EACH ROW
28 DECLARE
29 vdepocli deposito.cod_depo%TYPE ;
30 vqtde_estoque armazenamento.qtde_estoque%TYPE ;
31 BEGIN
32 -- buscando o deposito do cliente que fez o pedido
33 SELECT d.cod_depo INTO vdepocli
34 FROM deposito d, regiao r, cliente c, pedido p
35 WHERE d.cod_regiao = r.cod_regiao
36 AND r.cod_regiao = c.cod_regiao
37 AND c.pais_cli = d.pais_depo
38 AND p.cod_cli = c.cod_cli
39 AND p.num_ped = :NEW.num_ped ;
40 -- buscando a qtde de estoque para o produto armazenado no deposito acima
41 SELECT a.qtde_estoque INTO vqtde_estoque
42 FROM armazenamento a
43 WHERE a.cod_depo = vdepocli
44 AND a.cod_prod = :NEW.cod_prod ;
45 -- validar se a qtde é suficiente
46 IF vqtde_estoque < :NEW.qtde_pedida THEN
47 RAISE_APPLICATION_ERROR ( -20003,
48 'Qtde insuficiente para o item. Faltam '||
49 TO_CHAR (:NEW.qtde_pedida - vqtde_estoque)|| ' unidades');
50 END IF ;
51 END ;
52
53 SELECT cod_cli FROM pedido WHERE num_ped = 1900 ;
54 SELECT cod_regiao, pais_cli FROM cliente WHERE cod_cli = 200 ;
55 SELECT cod_depo FROM deposito WHERE cod_regiao = 2 AND pais_depo = 'BRA' ; -- 201
56 SELECT * FROM itens_pedido WHERE num_ped = 1900 ; -- 5005 tem 4 pedido
57 SELECT qtde_estoque FROM armazenamento WHERE cod_depo = 101 AND cod_prod = 5002 ;
58
59 SELECT * FROM armazenamento WHERE cod_depo = 101 ;
60 SELECT cod_regiao FROM deposito WHERE cod_depo = 101 ;
61 SELECT cod_cli FROM cliente WHERE cod_regiao = 1 ;
62 SELECT num_ped from pedido WHERE cod_cli = 203 ;
63 SELECT * FROM itens_pedido WHERE num_ped = 1901 ;
64 SELECT * FROM pedido WHERE num_ped = 1901 ;
65 SELECT cod_regiao
66
67 UPDATE itens_pedido SET qtde_pedida = 181 WHERE num_ped = 1910
68 AND cod_prod = 5002 ;
69 SELECT * FROM itens_pedido WHERE num_ped = 1910 AND cod_prod = 5002 ;
70
71 SELECT qtde_estoque
72 cod_regiao = 5 ;
73 SELECT cod_regiao, pais_cli FROM cliente WHERE cod_cli = 201 ;
74 SELECT * from deposito where cod_regiao = 5 ;
75
76 SELECT * FROM armazenamento WHERE cod_prod = 5002 ;
77 SELECT cod_regiao FROM deposito WHERE cod_depo IN ( 101, 105 ) ;
78 SELECT cod_cli FROM cliente WHERE cod_regiao = 6 ;
79 SELECT * FROM pedido WHERE cod_cli = 210 ;
80 SELECT * FROM itens_pedido WHERE num_ped = 1910 ;
81
82 SELECT * FROM pedido where cod_cli = 210 ; 1910
83 SELECT * FROM cliente WHERE cod_regiao = 6 ; 207 ou 210
84 SELECT * FROM itens_pedido WHERE num_ped = 1910 ;
85 SELECT * FROM deposito ; dep=105 , r=6 , pais = ALE
86 SELECT * FROM armazenamento where cod_depo = 105 ; prod = 5002 tem 150
87
88 update itens_pedido set qtde_pedida = 151 where cod_prod = 5002 AND num_ped = 1910 ;
89
90 /*****
91 Gatilho para atualizar o valor do pedido conforme inserção ou alteração no item do pedido - FATEC 21/fev/2019 -- OK

```

```

92 *****/
93 ALTER TABLE itens_pedido ADD situacao_item CHAR(15)
94 CHECK ( situacao_item IN ( 'SEPARACAO', 'ENTREGUE', 'CANCELADO', 'DESPACHADO' ) ) ;
95
96 UPDATE itens_pedido SET situacao_item = 'SEPARACAO' ;
97 ALTER TRIGGER valida_estoque ENABLE ;
98
99 CREATE OR REPLACE TRIGGER atualiza_pedido
100 AFTER INSERT OR UPDATE ON itens_pedido
101 FOR EACH ROW
102 DECLARE
103 vtotalped_antes pedido.vl_total_ped%TYPE ;
104 vtotalped_depois pedido.vl_total_ped%TYPE ;
105 vpreco_prod produto.preco_venda%TYPE ;
106 vpedido pedido.num_ped%TYPE ;
107 BEGIN
108 -- busca o preço do produto que está sendo manipulado
109 SELECT pr.preco_venda INTO vpreco_prod
110 FROM produto pr
111 WHERE pr.cod_prod = :NEW.cod_prod ;
112 -- no caso de insercao somar o valor do item
113 IF INSERTING THEN
114 SELECT p.vl_total_ped INTO vtotalped_antes
115 FROM pedido p
116 WHERE p.num_ped = :NEW.num_ped ;
117 vpedido := :NEW.num_ped ;
118 vtotalped_depois := vtotalped_antes +
119 :NEW.qtde_pedida * vpreco_prod * ( 100 - :NEW.descto_itens_pedido)/100;
120 ELSIF UPDATING AND :OLD.situacao_item = 'SEPARACAO' THEN
121 SELECT p.vl_total_ped INTO vtotalped_antes
122 FROM pedido p
123 WHERE p.num_ped = :OLD.num_ped ;
124 vpedido := :OLD.num_ped ;
125 -- cancelamento - estornar o valor do item
126 IF :NEW.situacao_item = 'CANCELADO' AND
127 :NEW.situacao_item <> :OLD.situacao_item THEN
128 vtotalped_depois := vtotalped_antes - :OLD.qtde_pedida*vpreco_prod*( 100 - :OLD.descto_itens_pedido)/ 100 ;
129 END IF ;
130 -- mudando a qtde, mas desconto igual
131 IF :NEW.qtde_pedida <> :OLD.qtde_pedida AND :NEW.descto_itens_pedido = :OLD.descto_itens_pedido THEN
132 vtotalped_depois := vtotalped_antes + (:NEW.qtde_pedida - :OLD.qtde_pedida)*vpreco_prod*
133 ( 100 - :OLD.descto_itens_pedido)/100 ;
134 END IF ;
135 -- mesma qtde, mudando desconto
136 IF :NEW.qtde_pedida = :OLD.qtde_pedida AND :NEW.descto_itens_pedido <> :OLD.descto_itens_pedido THEN
137 vtotalped_depois := vtotalped_antes + :OLD.qtde_pedida*vpreco_prod*
138 (:OLD.descto_itens_pedido-:NEW.descto_itens_pedido)/100 ;
139 END IF ;
140 -- muda qtde e desconto - valor antes + diferença novo valor total item e valor total item antes
141 IF :NEW.qtde_pedida <> :OLD.qtde_pedida AND :NEW.descto_itens_pedido <> :OLD.descto_itens_pedido THEN
142 vtotalped_depois := vtotalped_antes + :NEW.qtde_pedida*vpreco_prod* ( 100 -:NEW.descto_itens_pedido)/100 -
143 :OLD.qtde_pedida*vpreco_prod* ( 100 -:OLD.descto_itens_pedido)/100;
144 END IF ;
145 END IF ;
146 -- atualiza o valor na tabela pedido
147 IF vtotalped_antes <> vtotalped_depois THEN
148 UPDATE pedido p SET p.vl_total_ped = vtotalped_depois
149 WHERE p.num_ped = vpedido ;
150 END IF ;
151 END ;
152
153 ALTER TRIGGER valida_estoque DISABLE ;
154
155 SELECT * FROM pedido WHERE num_ped = 1905 ;
156 SELECT * FROM itens_pedido WHERE num_ped = 1905 ;
157 SELECT * FROM produto WHERE cod_prod = 5003 ;
158 UPDATE produto SET preco_venda = 100 WHERE cod_prod = 5003 ; -- mudando preço para 100
159 UPDATE ITENS_pedido SET DESCTO_ITENS_PEDIDO = 10 WHERE num_ped = 1905 ; -- todos itens com 10%
160 -- recalculando o valor total dos pedidos
161 UPDATE pedido ped
162 SET ped.vl_total_ped =
163 (SELECT sum(i.qtde_pedida*p.preco_venda*(100-i.descto_itens_pedido)/100)
164 FROM itens_pedido i, produto p
165 WHERE ped.num_ped = i.num_ped
166 AND i.cod_prod = p.cod_prod );
167
168 -- Testes
169
170 -- Pedido 1905 -> $ 12024 -- todos itens com 10% desconto e item 5003 $100
171 -- mudando a qtde -- aumentei 10, era 30 foi pra 40
172 UPDATE itens_pedido SET QTDE_PEDIDA = 40 WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; -- 12924
173 -- 10x(100-10)/100 = 900 a mais // bateu
174 -- voltando -- colocando 30 de novo
175 UPDATE itens_pedido SET QTDE_PEDIDA = 30 WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; -- 12024 // bateu
176 -- mudando desconto -- mais 10% -- preço passou a 80, diminui 10 reais por unidade x 30 unidades diminui 300
177 UPDATE itens_pedido SET DESCTO_ITENS_PEDIDO = 20 WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; --11724//bateu
178 -- voltando
179 UPDATE itens_pedido SET DESCTO_ITENS_PEDIDO = 10 WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; --12024//bateu
180 -- mudando os dois, unidade custa 100, aumentou 10 unidades pedidas então
181 -- $100 x 40 - $90x30 = 1300 a mais
182 UPDATE itens_pedido SET QTDE_PEDIDA = 40, DESCTO_ITENS_PEDIDO = 0
183 WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; -- 13324 // bateu

```

```
184  -- voltando
185  UPDATE itens_pedido SET QTDE_PEDIDA = 30, DESCTO_ITENS_PEDIDO = 10
186  WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; -- 12024 //bateu
187  -- mudando os dois, unidade custa 80 agora , diminui 10 unidades então
188  -- $90x30 - $80x20 = 2700 - 1600 -> diminui 1100
189  UPDATE itens_pedido SET QTDE_PEDIDA = 20, DESCTO_ITENS_PEDIDO = 20
190  WHERE COD_PROD = 5003 AND NUM_PED = 1905 ; -- 10924 = 12024 - 1100 //bateu
```