

```

1  /*****
2  F U N Ç Õ E S - Aula 28/fev/2019
3  *****/
4  --Elaborar uma função que retorne o preço do produto passando seu código como parâmetro ;
5  CREATE OR REPLACE FUNCTION get_preco ( vprod IN produto.cod_prod%TYPE )
6  RETURN NUMBER IS
7  vpreco produto.preco_venda%TYPE ;
8  BEGIN
9  SELECT preco_venda INTO vpreco
10 FROM produto
11 WHERE cod_prod = vprod ;
12 RETURN vpreco ;
13 EXCEPTION
14 WHEN NO_DATA_FOUND THEN
15 RAISE_APPLICATION_ERROR ( -20999, 'Produto '||TO_CHAR(vprod)||' não localizado!!!');
16 END ;
17 -- testando
18 SELECT get_preco( 5050) FROM dual ;
19
20 /* Aula FATEC 28/fev/2019 Elabore uma função que retorne o preço sugerido de um produto passando como parâmetro o nome ou parte do nome.
21 Suponha que o nome do produto é único, não se repete. */
22 CREATE OR REPLACE FUNCTION get_preco_nome ( vprod IN produto.nome_Prod%TYPE)
23 RETURN NUMBER IS
24 vpreco produto.preco_venda%TYPE ;
25 vbusca produto.nome_prod%TYPE := '%'||UPPER(vprod)||'%';
26 vqtos_tem SMALLINT := 0 ;
27 BEGIN
28 SELECT COUNT(*) INTO vqtos_tem
29 FROM produto
30 WHERE UPPER(nome_prod) LIKE vbusca ;
31 IF vqtos_tem = 0 THEN
32 RAISE_APPLICATION_ERROR ( -20100, 'Produto não localizado !!!') ;
33 ELSIF vqtos_tem > 1 THEN
34 RAISE_APPLICATION_ERROR ( -20101, 'Mais de um produto com este nome! Refine sua busca!') ;
35 ELSIF vqtos_tem = 1 THEN
36 SELECT preco_venda INTO vpreco
37 FROM produto
38 WHERE UPPER(nome_prod) LIKE vbusca ;
39 END IF;
40 RETURN vpreco ;
41 END ;
42 -- testando
43 SELECT get_preco_nome ( 'blablabla') FROM dual ;
44 SELECT get_preco_nome ( 'tenis') FROM dual ;
45 SELECT get_preco_nome ( 'tenis slim') FROM dual ;
46
47
48 /*****
49 Atividades 3 e 4 - Fatec Correção 07/mar/2019
50 *****/
51 --3-Validar o vendedor que atende o pedido garantindo que o cargo deste funcionário corresponde ao de um Vendedor;
52 CREATE OR REPLACE TRIGGER validavendedor
53 BEFORE INSERT OR UPDATE ON pedido
54 FOR EACH ROW
55 DECLARE
56 vcargo cargo.nome_cargo%TYPE ;
57 BEGIN
58 select nome_cargo into vcargo
59 from funcionario f, cargo c
60 where f.cod_cargo = c.cod_cargo
61 and f.reg_func = :NEW.reg_func_vendedor ;
62 IF upper(vcargo) NOT LIKE '%VENDEDOR%' THEN
63 RAISE_APPLICATION_ERROR ( -20003, 'Vendedor não confere com o cargo!!. Ele é '||vcargo);
64 END IF ;
65 END ;
66
67 select * from funcionario ;
68 select * from cargo ;
69 SELECT * from pedido ;
70
71 INSERT INTO pedido VALUES ( pedido_cod.nextval, current_timestamp - 100, 'FONE' , 1000, 15, 12, 'o mesmo',
72 'CTCREd' , 200, 15 , 'APROVADO'); -- ok deu erro
73
74 UPDATE pedido SET reg_func_vendedor = 1 WHERE num_ped = 1920 ;
75
76 --4-Validar a região do vendedor que atende o pedido para a mesma região do cliente, ou seja,
77 --não permitir que um vendedor de outra região atenda o cliente;
78 descr pedido ;
79 descr funcionario ;
80 descr cliente ;
81 descr regioao;
82 SELECT * FROM cargo ;
83
84 CREATE OR REPLACE trigger valida_vendedor_regiao
85 BEFORE INSERT OR UPDATE ON pedido
86 FOR EACH ROW
87 DECLARE
88 rvendedor regioao.cod_regiao%TYPE ; -- captura a regioao do vendedor
89 rcliente regioao.cod_regiao%TYPE ; -- captura a regioao do cliente
90 BEGIN
91 SELECT c.cod_regiao INTO rcliente
92 from cliente c
93 where c.cod_cli =:NEW.cod_cli ;
94
95 SELECT f.cod_regiao INTO rvendedor
96 from funcionario f
97 where f.reg_func = :NEW.reg_func_vendedor ;
98
99 IF rcliente <> rvendedor THEN
100 RAISE_APPLICATION_ERROR ( -20004 , 'Vendedor nao pertence à mesma regioao do cliente !!'
101 ||'Vendedor : '||TO_CHAR(rvendedor)|| ' x Cliente : '||TO_CHAR(rcliente));
102 END IF ;
103 END ;
104
105 descr pedido ;
106 INSERT INTO pedido VALUES ( pedido_cod.nextval, current_timestamp - 100, 'FONE' , 1000, 15, 12, 'o mesmo',
107 'CTCREd' , 200, 4 , 'APROVADO'); -- ok erro

```

