

1) Crie um procedimento chamado prc\_inserir\_produto para todas as colunas da tabela de produtos, valide:

Se o nome do produto tem mais de 3 caracteres e não contém números (0 a 9).

```
CREATE OR REPLACE PROCEDURE PRC_INSERE_PRODUTO (P_COD_PRODUTO
PRODUTO.COD_PRODUTO%TYPE,
                                                    P_NOM_PRODUTO
PRODUTO.NOM_PRODUTO%TYPE,
                                                    P_COD_BARRA
PRODUTO.COD_BARRA%TYPE,
                                                    P_STA_ATIVO
PRODUTO.STA_ATIVO%TYPE,
                                                    P_DAT_CADASTRO
PRODUTO.DAT_CADASTRO%type,
                                                    P_DAT_CANCELAMENTO
PRODUTO.DAT_CANCELAMENTO%TYPE) IS
BEGIN
    IF LENGTH(P_NOM_PRODUTO) <= 3 or REGEXP_LIKE(P_NOM_PRODUTO, '[0-9]') then
        RAISE_APPLICATION_ERROR(-20003,
                                'Nome do produto deve ter mais de 3 caracteres e
não possuir números ');
    end if;
    INSERT INTO PRODUTO
        (cod_produto,
         nom_produto,
         cod_barra,
         sta_ativo,
         dat_cadastro,
         dat_cancelamento)
    VALUES
        (P_cod_produto,
         P_nom_produto,
         P_cod_barra,
         P_sta_ativo,
         P_dat_cadastro,
         P_dat_cancelamento);

    COMMIT;

END PRC_INSERE_PRODUTO;
```

execute PRC\_INSERE\_PRODUTO(12347,'Garrafa de agua 5','12223555','S',sysdate,sysdate+10);

2 Crie um procedimento chamado prc\_inserir\_cliente para inserir novos clientes, valide:

Se o nome do cliente tem mais de 3 caracteres e não contém números (0 a 9)

```
CREATE OR REPLACE PROCEDURE prc_inserir_cliente(p_cod_cliente
clientes.cod_cliente%type,
                                                    p_nom_cliente
clientes.nom_cliente%type,
                                                    p_des_razao_social
clientes.des_razao_social%type,
                                                    p_tip_pessoa
clientes.tip_pessoa%type,
                                                    p_num_cpf_cnpj
clientes.num_cpf_cnpj%type,
                                                    p_dat_cadastro
clientes.dat_cadastro%type,
                                                    p_dat_cancelamento
clientes.dat_cancelamento%type,
                                                    p_sta_ativo
clientes.sta_ativo%type) IS
BEGIN
    -- Validação do nome do cliente
    IF LENGTH(p_nom_cliente) <= 3 OR REGEXP_LIKE(p_nom_cliente, '[0-9]') THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nome do cliente inválido.');
```

END IF;

-- Inserção do novo cliente

```
INSERT INTO clientes
(cod_cliente,
 nom_cliente,
 des_razao_social,
 tip_pessoa,
 num_cpf_cnpj,
 dat_cadastro,
 dat_cancelamento,
 sta_ativo)
VALUES
(p_cod_cliente,
 p_nom_cliente,
 p_des_razao_social,
 p_tip_pessoa,
 p_num_cpf_cnpj,
 p_dat_cadastro,
 p_dat_cancelamento,
 p_sta_ativo);

-- Confirmação da transação
COMMIT;
```

END prc\_inserir\_cliente;

Execute prc\_inserir\_cliente(1233,'Maria dos Santos','descrição','F',78925632515,sysdate,sysdate+15,'s');

- 3 Crie uma função chamada FUN\_VALIDA\_NOME que valide se o nome tem mais do que 3 caracteres e não tenha números.

```
CREATE OR REPLACE FUNCTION FUN_VALIDA_NOME2(p_nome IN VARCHAR2) RETURN
BOOLEAN IS
BEGIN
    IF LENGTH(p_nome) > 3 AND NOT REGEXP_LIKE(p_nome, '[0-9]') THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;
END;
```

```
CREATE OR REPLACE FUNCTION FUN_VALIDA_NOME(p_nome IN VARCHAR2) RETURN
VARCHAR2 IS
BEGIN
    IF LENGTH(p_nome) > 3 AND NOT REGEXP_LIKE(p_nome, '[0-9]') THEN
        RETURN 'Nome válido.';
    ELSE
        RETURN 'Nome inválido.';
    END IF;
END;
```

```
SELECT FUN_VALIDA_NOME ('VERGILIO SANTOS') FROM DUAL;
```

4. Altere os procedimentos dos exercícios 1 e 2 para chamar a função do exercício 3.

```
CREATE OR REPLACE PROCEDURE prc_insere_cliente2(p_cod_cliente
cliente.cod_cliente%type,
                                                    p_nom_cliente
cliente.nom_cliente%type,
                                                    p_des_razao_social
cliente.des_razao_social%type,
                                                    p_tip_pessoa
cliente.tip_pessoa%type,
                                                    p_num_cpf_cnpj
cliente.num_cpf_cnpj%type,
                                                    p_dat_cadastro
cliente.dat_cadastro%type,
                                                    p_dat_cancelamento
cliente.dat_cancelamento%type,
                                                    p_sta_ativo
cliente.sta_ativo%type) IS
BEGIN
    -- Validação do nome do cliente
    IF FUN_VALIDA_NOME2(p_nom_cliente) = false THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nome do cliente inválido.');
```

```

-- Inserção do novo cliente
INSERT INTO cliente
(cod_cliente,
 nom_cliente,
 des_razao_social,
 tip_pessoa,
 num_cpf_cnpj,
 dat_cadastro,
 dat_cancelamento,
 sta_ativo)
VALUES
(p_cod_cliente,
 p_nom_cliente,
 p_des_razao_social,
 p_tip_pessoa,
 p_num_cpf_cnpj,
 p_dat_cadastro,
 p_dat_cancelamento,
 p_sta_ativo);

-- Confirmação da transação
COMMIT;

END prc_insere_cliente;

```

5. Altere o procedimento do exercício 1 para que tenha um último parâmetro chamado P\_RETORNO do tipo varchar2 que deverá retornar à informação 'produto cadastrado com sucesso'.

```

CREATE OR REPLACE PROCEDURE PRC_INSERE_PRODUTO2 (P_COD_PRODUTO
PRODUTO.COD_PRODUTO%TYPE,
                                                    P_NOM_PRODUTO
PRODUTO.NOM_PRODUTO%TYPE,
                                                    P_COD_BARRA
PRODUTO.COD_BARRA%TYPE,
                                                    P_STA_ATIVO
PRODUTO.STA_ATIVO%TYPE,
                                                    P_DAT_CADASTRO
PRODUTO.DAT_CADASTRO%TYPE,
                                                    P_DAT_CANCELAMENTO
PRODUTO.DAT_CANCELAMENTO%TYPE,
                                                    p_MENSAGEN_RETORNO OUT
VARCHAR2) IS

BEGIN
  IF LENGTH(P_NOM_PRODUTO) <= 3 or REGEXP_LIKE(P_NOM_PRODUTO, '[0-9]') then
    p_MENSAGEN_RETORNO := 'Nome do produto inválido';
  else
    INSERT INTO PRODUTO
      (cod_produto,
       nom_produto,
       cod_barra,
       sta_ativo,

```

```

        dat_cadastro,
        dat_cancelamento)
VALUES
(P_cod_produto,
 P_nom_produto,
 P_cod_barra,
 P_sta_ativo,
 P_dat_cadastro,
 P_dat_cancelamento);

p_MENSAGEN_RETORNO := 'Produto inserido com sucesso.';

end if;

COMMIT;

END PRC_INSERE_PRODUTO2;

```

6. crie um bloco anônimo e chame o procedimento do exercicio 1.

```

DECLARE
    p_MENSAGEN_RETORNO VARCHAR2(100);
BEGIN
    prc_inserir_produto2(5566, 'Garrafa de agua
5', '12223555', 'S', sysdate, sysdate+10, p_MENSAGEN_RETORNO);
    DBMS_OUTPUT.PUT_LINE(p_MENSAGEN_RETORNO);
END;
/

DECLARE
    p_MENSAGEN_RETORNO VARCHAR2(100);
BEGIN
    prc_inserir_produto2(5566, 'Garrafa de agua
', '12223555', 'S', sysdate, sysdate+10, p_MENSAGEN_RETORNO);
    DBMS_OUTPUT.PUT_LINE(p_MENSAGEN_RETORNO);
END;
/

```