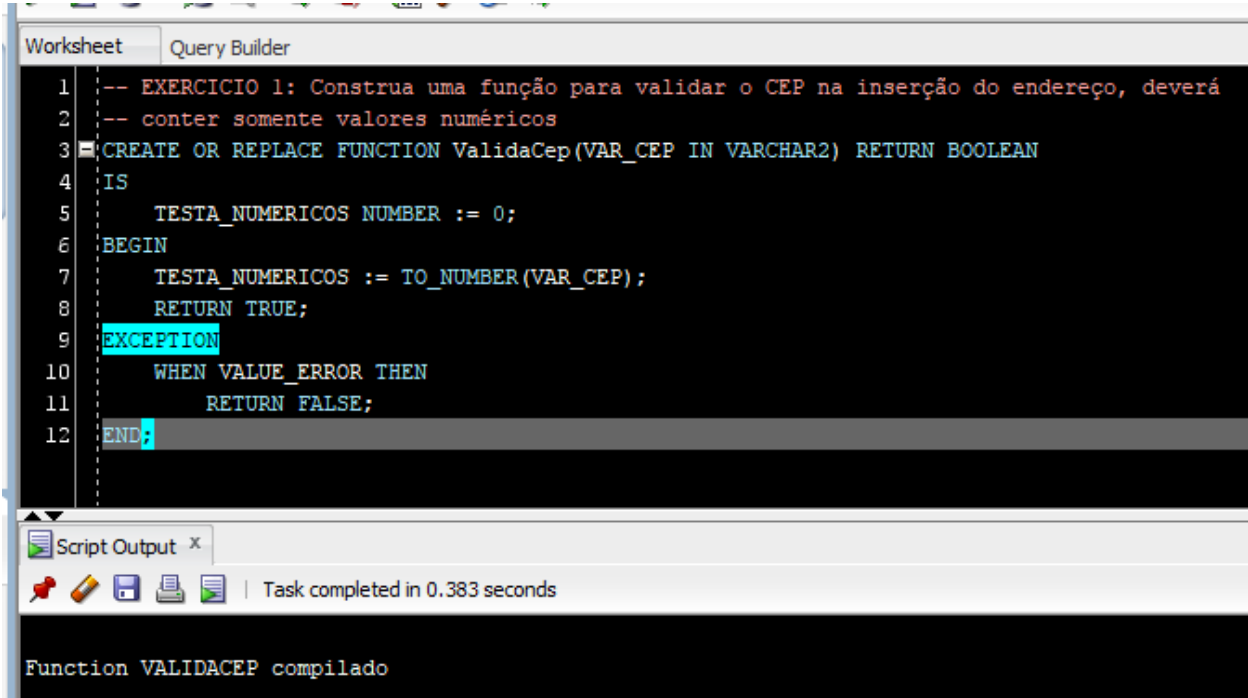


## Exercícios

1 Construa uma função para validar o CEP na inserção do endereço, deverá conter somente valores numéricos.



```
1  -- EXERCICIO 1: Construa uma função para validar o CEP na inserção do endereço, deverá
2  -- conter somente valores numéricos
3  CREATE OR REPLACE FUNCTION ValidaCep(VAR_CEP IN VARCHAR2) RETURN BOOLEAN
4  IS
5      TESTA_NUMERICOS NUMBER := 0;
6  BEGIN
7      TESTA_NUMERICOS := TO_NUMBER(VAR_CEP);
8      RETURN TRUE;
9  EXCEPTION
10     WHEN VALUE_ERROR THEN
11         RETURN FALSE;
12 END;
```

Script Output x

Task completed in 0.383 seconds

Function VALIDACEP compilado

2 – Crie uma procedure para inserir novos clientes, todos os valores deverão ser informados por parâmetros e deverá ter tratamento de exceção.

Worksheet Query Builder

```
1  -- EXERCICIO 2: Crie uma procedure para inserir novos clientes, todos os valores deverão ser informados por p
2  -- e deverá ter tratamento de exceção
3  CREATE OR REPLACE PROCEDURE InserirCliente(
4      CODIGO CLIENTE.COD_CLIENTE%TYPE,
5      NOME CLIENTE.NOM_CLIENTE%TYPE,
6      RAZAO_SOCIAL CLIENTE.DES_RAZAO_SOCIAL%TYPE,
7      TIPO CLIENTE.TIP_PESSOA%TYPE,
8      CPF_CNPJ CLIENTE.NUM_CPF_CNPJ%TYPE,
9      CADASTRO CLIENTE.DAT_CADASTRO%TYPE,
10     CANCELAMENTO CLIENTE.DAT_CANCELAMENTO%TYPE,
11     STATUS CLIENTE.STA_ATIVO%TYPE
12 ) IS
13     DML_ERRORS EXCEPTION;
14     PRAGMA EXCEPTION_INIT(DML_ERRORS, -24381);
15 BEGIN
16     INSERT INTO CLIENTE VALUES (CODIGO, NOME, RAZAO_SOCIAL, TIPO, CPF_CNPJ, CADASTRO, CANCELAMENTO, STATUS);
17     COMMIT;
18     EXCEPTION
19     WHEN DML_ERRORS THEN
20         DBMS_OUTPUT.PUT_LINE('Dados recebidos incorretamente');
21         ROLLBACK;
22 END;
```

Script Output x

Task completed in 5.26 seconds

Procedure INSERIRCLIENTE compilado

3 Crie um procedure que para atualizar a quantidade de produtos e estoque.

Worksheet Query Builder

```
1  -- EXERCICIO 3: Crie uma procedure que atualize a quantidade de produtos e estoque
2  CREATE OR REPLACE PROCEDURE AttProdutoEstoque(
3      COD_PRODUTO ESTOQUE_PRODUTO.COD_PRODUTO%TYPE,
4      QTD_PRODUTO_DESEJADO ESTOQUE_PRODUTO.QTD_PRODUTO%TYPE,
5      ESTOQUE_ESTOQUE_PRODUTO.COD_ESTOQUE%TYPE
6  ) IS
7  BEGIN
8      UPDATE ESTOQUE_PRODUTO SET COD_ESTOQUE = ESTOQUE, QTD_PRODUTO = QTD_PRODUTO_DESEJADO WHERE COD_PRODUTO = COD_PRODUTO;
9      COMMIT;
10 END;
```

Script Output x

Task completed in 10.183 seconds

Procedure ATTPRODUTOESTOQUE compilado

4 – Crie uma procedure para deletar um pedido.

Worksheet | Query Builder

1

-- EXERCICIO 4: CRIE UMA PROCEDURE PARA DELETAR UM PEDIDO

2

CREATE OR REPLACE PROCEDURE DeletaPedido(

3

CODIGO PEDIDO.COD\_PEDIDO%TYPE

4

) IS

5

BEGIN

6

DELETE FROM ITEM\_PEDIDO WHERE COD\_PEDIDO = CODIGO;

7

DELETE FROM HISTORICO\_PEDIDO WHERE COD\_PEDIDO = CODIGO;

8

DELETE FROM PEDIDO WHERE COD\_PEDIDO = CODIGO;






9

COMMIT;

10

END;

Script Output x

     | Task completed in 5.189 seconds

Procedure DELETAPEDIDO compilado