

### Problema 1:

1) Executar o comando abaixo e explicar o que ele faz:

```
DECLARE
    a number;
    b number;
    c number;

PROCEDURE valorMinimo(x IN number, y IN number, z OUT number)
IS
BEGIN
    IF x < y THEN
        z:= x;
    ELSE
        z:= y;
    END IF;
END;

BEGIN
    a:= 23;
    b:= 45;
    valorMinimo(a, b, c);
    dbms_output.put_line('O valor mínimo obtido entre os
valores ' || a || ' e ' || b || ' foi ' || c);
END;
```

### **RESPOSTA:**

**A Procedure recebe de entrada os valores de x e y e devolve na saída o valor z. Depois, compara o x e y para ver qual é o menor valor e atribui o mesmo a z.**

2) Você conseguiria fazer um comando semelhante, mas agora para identificar o valor máximo e o valor médio?

### **RESPOSTA:**

**//Valor máximo**

```
DECLARE
    a number;
    b number;
    c number;

PROCEDURE valorMaximo(x IN number, y IN number, z OUT number)
IS
BEGIN
    IF x > y THEN
        z:= x;
    ELSE
        z:= y;
    END IF;
```

```

END;

BEGIN
    a:= 23;
    b:= 45;
    valorMinimo(a, b, c);
    dbms_output.put_line('O valor máximo obtido entre os
valores ' || a || ' e ' || b || ' foi ' || c);
END;

//Valor médio

DECLARE
    a number;
    b number;
    c number;

PROCEDURE valorMedio(x IN number, y IN number, z OUT number)
IS
BEGIN
    z := (x+y)/2;
END;

BEGIN
    a:= 23;
    b:= 45;
    valorMedio(a, b, c);
    dbms_output.put_line('O valor medio obtido entre os valores
' || a || ' e ' || b || ' foi ' || c);
END;

```

## Problema 2:

1) Criar a seguinte tabela:

```

CREATE TABLE CIDADES
(
    COD_CIDADE    INTEGER NOT NULL,
    NOME           VARCHAR2(100 BYTE),
    UF             CHAR(2 BYTE)
);

```

2) Executar o comando abaixo:

```

CREATE OR REPLACE PROCEDURE "SP_CIDADES" (
    vCOD_CIDADE    INTEGER,
    vNOME           VARCHAR2,
    vUF             CHAR,
    vOPR            CHAR)
IS
    vEXCEPTION EXCEPTION;

```

```

BEGIN
  IF (vOPR = 'I') THEN
    INSERT INTO CIDADES(COD_CIDADE, NOME, UF)
      VALUES (vCOD_CIDADE, vNOME, vUF);
  ELSE

    IF(vOPR = 'A') THEN
      UPDATE CIDADES
        SET NOME = vNOME, UF = vUF WHERE COD_CIDADE = vCOD_CIDADE;

    ELSE
      IF(vOPR = 'D') THEN
        DELETE FROM CIDADES WHERE COD_CIDADE = vCOD_CIDADE;
      ELSE
        RAISE vEXCEPTION;
      END IF;
    END IF;
  END IF;
END IF;
EXCEPTION
  WHEN vEXCEPTION THEN
    RAISE_APPLICATION_ERROR(-20999,'ATENÇÃO! Operação diferente
de Inserção, Deleção ou Atualização.', FALSE);
END SP_CIDADES;

```

3) Para observar os resultados, execute o comando abaixo:

```
EXEC SP_CIDADES(1, 'GASPAR', 'SC', 'I');
```

4) Tente executar o mesmo comando anterior sem informar algum dado ou informando tipos de dados diferentes. Relate o que acontece.

#### RESPOSTA:

O único campo que gera algum erro é o último (VOPR), pois está previsto na exceção. Basicamente, quando é um 'I' fará uma inserção, quando é um 'A' fará um update, quando é um 'D' fará uma deleção. Então, ao não preencher nada ou alguma outra letra, dispara um erro de aplicação "ATENÇÃO! Operação diferente de Inserção, Deleção ou Atualização."

Problema 3:

1) Criar a seguinte tabela e explicar o que aconteceu:

```
create table funcionarios as select * from hr.employees;
```

**RESPOSTA:**

**Copia as linhas e colunas da tabela employees para a tabela funcionários.**

2) Executar o comando abaixo:

```
CREATE OR REPLACE PROCEDURE detalhes_dos_funcionarios
IS
    CURSOR emp_cur IS
        SELECT first_name, last_name, salary
        FROM funcionarios;
    emp_rec emp_cur%rowtype;
BEGIN
    FOR emp_rec in emp_cur LOOP
        dbms_output.put_line('Nome do funcionário: ' ||
emp_rec.first_name ||
        '. Sobrenome do funcionário: ' || emp_rec.last_name ||
        '. Salário do funcionário: ' || emp_rec.salary);
    END LOOP;
END;
```

3) Para observar os resultados, execute o comando abaixo e explique o que o comando fez:

```
exec detalhes_dos_funcionarios;
```

**RESPOSTA:**

**Traz uma linha para cada funcionário, informando o nome, sobrenome e salário presentes na tabela, que copiou as informações da tabela employees.**

#### Problema 4:

1) Executar o comando abaixo:

```
CREATE OR REPLACE FUNCTION PRIMO (NUMERO NUMBER) RETURN
VARCHAR2
IS
    VSQRT NUMBER(4);
    VDIV NUMBER(4);
    VRESULT VARCHAR(20) := 'É PRIMO';

BEGIN
    --RAIZ QUADRADA DO NUMERO
    VSQRT := SQRT(NUMERO);

    FOR I IN 2..VSQRT LOOP
        IF (MOD(NUMERO,I)=0 AND NUMERO<>I) THEN
            VRESULT := 'NÃO É PRIMO';
        END IF;
    END LOOP;
    RETURN VRESULT;
END;
```

2) Para observar os resultados, execute o comando abaixo e explique o que o comando fez:

```
BEGIN
    DBMS_OUTPUT.PUT_LINE(PRIMO(4));
END;
```

#### **RESPOSTA:**

**No comando, é passado o valor "4" para ser realizada a validação de se é primo ou não (não é).**

3) Consegue perceber a diferença do tipo de comando deste problema se comparado aos problemas 2 e 3? Explique.

#### **RESPOSTA:**

**É o único que usou function.**