

QUESTÕES – AVALIAÇÃO A1

Para cada cursor explícito, quatro atributos podem ser verificados e seus valores alterados durante a execução do código. Esses atributos são:

- a) Found, NotFound, RowCount e IsOpen
- b) Before, After, Found e NotFound
- c) In, Out, Found e NotFound
- d) In, Out, RowCount e IsOpen
- e) In, Out, Before e After

Analisar o programa abaixo e verificar qual será o valor final da variável V_PASSO.

```
DECLARE
    V_PASSO NUMBER(4) := 4;
BEGIN
    V_PASSO := V_PASSO + 1;
    LOOP
        EXIT WHEN V_PASSO > 25;
        V_PASSO := (V_PASSO * V_PASSO) - 8;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('O valor da variável é: ' || V_PASSO);
END;
```

- a) 281
- b) 262
- c) 5
- d) 145
- e) 17

Analisando o bloco PL/SQL abaixo, em que circunstância o programa passará pela área EXCEPTION:

```
DECLARE
    V_CONTA    CONTA.NUMCONTA%TYPE;
    V_MSG      CONTA.OBSERV%TYPE;
BEGIN
    SELECT NUMCONTA INTO V_CONTA FROM CONTA WHERE SALDO < 10000;
    V_MSG := 'Saldo da conta nº ' || V_CONTA || ' baixo';
    INSERT INTO HISTCONTA
    VALUES
    ('N', sysdate, V_MSG);
EXCEPTION
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Muitas contas com saldo inferior a 10000!');
END;
```

- a) Quando o comando SELECT retornar mais de um registro.
- b) Quando o comando INSERT finalizar a inserção com sucesso.
- c) Quando o comando SELECT não encontrar registro.
- d) Quando o comando INSERT não conseguir inserir registro.
- e) Sempre que o comando SELECT encontrar registro.

Analise o programa abaixo e escolha a alternativa que represente a mensagem que será mostrada na tela se os seguintes valores forem informados em tempo de execução para as variáveis: a = 5, b = 6, c = 12.

```
DECLARE
  a number(2) := &a;
  b number(2) := &b;
  c number(2) := &c;
  v_mens varchar(40);
BEGIN
  IF ( a < b + c ) AND ( b < a + c ) AND ( c < a + b ) THEN
    IF ( a = b ) AND ( b = c ) THEN
      v_mens := 'Triângulo Equilátero';
    ELSE
      IF ( a = b ) OR ( b = c ) OR ( a = c ) THEN
        v_mens := 'Triângulo Isósceles';
      ELSE
        v_mens := 'Triângulo Escaleno';
      END IF;
    END IF;
  ELSE
    v_mens := 'Não é possível formar um triângulo';
  END IF;
  DBMS_OUTPUT.PUT_LINE ( v_mens );
END;
```

- a) Não é possível formar um triângulo
- b) Triângulo Escaleno
- c) Triângulo Isósceles
- d) Triângulo Equilátero

Em relação ao bloco de declaração abaixo podemos afirmar que:

```
DECLARE
  CURSOR C_CLIENTE IS SELECT CODIGO, NOME
    FROM CLIENTE WHERE CIDADE = 'SANTOS'      ;
  V_CLIENTE C_CLIENTE%ROWTYPE                ;
  V_CPF      CLIENTE.CPF%TYPE                 ;
BEGIN
  .....                                       ;
  .....                                       ;
END;
```

- a) A variável v_cliente terá a estrutura correspondente aos campos codigo e nome do cursor c_cliente.
- b) A variável v_cliente terá a estrutura correspondente aos campos codigo, nome e cidade da tabela cliente.
- c) A variável v_cliente terá todos os clientes da cidade de Santos.
- d) A variável v_cliente terá a estrutura correspondente aos campos codigo, nome e cidade do cursor c_cliente.
- e) A variável v_cliente terá a mesma estrutura da tabela cliente (todos os campos).

Sobre a linguagem PL/SQL do Oracle, analise:

- I. A primeira seção de um bloco PL/SQL, chamada de declaração, é opcional. Contudo, se o bloco usar variáveis ou constantes, todas elas devem ser previamente declaradas antes de serem determinadas em um comando.
- II. Uma constante deve ser declarada na seção DECLARE, porém, sua inicialização pode ocorrer tanto na seção DECLARE quanto na seção BEGIN.
- III. O uso da cláusula INTO no comando SELECT, dentro de um programa PL/SQL, permite transferir o conteúdo dos campos de um registro de uma tabela para variáveis de memória.
- IV. Ao declarar uma variável utilizando o %TYPE, o mesmo herda o tipo e tamanho de um campo de uma tabela ou de uma variável definida anteriormente no código.

Está correto o que consta em:

- a) I, III e IV apenas
- b) II, III e IV apenas
- c) I, II, III e IV
- d) I, II e IV apenas
- e) I, II e III apenas

Analise o bloco PL_SQL. Quais serão os valores para as variáveis (x, y, z) durante toda a execução do programa?

```
DECLARE
  X number(3);
  Y number(3);
  Z number(3);
  N number(3) := 11;
BEGIN
  X := 18;
  LOOP
    WHILE N > 0 LOOP
      X := X + N;
      Z := X - N;
      N := N - 8;
      Y := N;
      DBMS_OUTPUT.PUT_LINE ('Valores para X, Y e Z são : ' || X || ', ' || Y || ', ' || Z);
    END LOOP;
    IF X > 18 THEN
      EXIT;
    END IF;
  END LOOP;
END;
```

- a) X = 29, Y = 3, Z = 18 e X = 32, Y = -5, Z = 29
- b) X = 29, Y = 4, Z = 18 e X = 33, Y = -3, Z = 29
- c) X = 28, Y = 2, Z = 18 e X = 30, Y = -6, Z = 28
- d) X = 29, Y = 3, Z = 18
- e) X = 30, Y = -6, Z = 28

Escolha a alternativa que complete as lacunas do código abaixo:

```

..... PROCEDURE Prod
(
  V_Cod      IN   PRODUTO.IDProd%Type ,
  V_Nome     IN   PRODUTO.Nome%Type   ,
  V_Val      IN   PRODUTO.Valor%Type
)
.....
BEGIN
  INSERT ..... Produto
  (IDProd, Nome, Valor)
  VALUES
  (V_Cod, V_Nome, V_Val)
  .....
  WHEN      DUP_VAL_ON_INDEX      THEN
    DBMS_OUTPUT.PUT_LINE('Produto já cadastrado!');
END;

```

- a) Create, IS, Into, Exception
- b) Open, IS, Into, Loop
- c) Create, To, Table, While
- d) Create, IS, To, Exception
- e) Open, To, Table, Exception

Escolha a alternativa que complete a função abaixo. A função deverá receber o número da matrícula de um funcionário e retornar seu nome.

```

CREATE or Replace Function NomeFunc
(
  P_Matric   IN   FUNC.IDMatric%Type
)
RETURN Varchar2
IS
  V_Nome     VARCHAR2(50)
BEGIN
  SELECT .....
  RETURN V_Nome
END NomeFunc;

```

- a) Select NOME into V_NOME from FUNC where IDMATRIC = P_MATRIC
- b) Select NOME in V_NOME from FUNC where IDMATRIC = V_NOME
- c) Select NOME in V_NOME from FUNC where IDMATRIC = P_MATRIC
- d) Select * into V_NOME from FUNC where IDMATRIC = P_MATRIC
- e) Select NOME into V_NOME from FUNC where IDMATRIC = V_NOME

Qual alternativa corresponde ao preenchimento correto das lacunas do código PL/SQL abaixo, cuja finalidade é apresentar o CNPJ e nome de todos os clientes da tabela CLIENTE cujo STATUS esteja como 'Ativo':

```
DECLARE
    ..... C_CLIENTE IS
        SELECT CNPJ, NOME FROM CLIENTE WHERE STATUS = 'ATIVO' ;
    V_REG      C_CLIENTE%ROWTYPE ;
BEGIN
    OPEN      C_CLIENTE ;
    FETCH     C_CLIENTE INTO V_REG ;
    ..... C_CLIENTE%FOUND LOOP
        DBMS_OUTPUT.PUT_LINE('CNPJ: ' || V_REG.CNPJ) ;
        DBMS_OUTPUT.PUT_LINE('NOME: ' || V_REG.NOME) ;
        FETCH C_CLIENTE INTO V_REG ;
    END ..... ;
    ..... C_CLIENTE
END;
```

- a) Cursor, While, Loop, Close
- b) Cursor, Rowtype, Loop, Open
- c) Cursor, For, Loop, Close
- d) Cursor, Rowtype, Loop, Close
- e) Cursor, For, Loop, Open