



1. Verifique o seguinte código:

```
Declare nome varchar2(40);
```

```
BEGIN
```

```
select nome_artista into nome from artista where cod_artista = 2;
```

```
dbms_output.put_line('Quantidade de registros retornados = ' || to_char(sql%ROWCOUNT));
```

```
dbms_output.put_line('Nome do artista = ' || nome);
```

```
END;
```

O que ele faz?? Quantos registros são retornados??

2. Verifique o seguinte código, explique o que ele faz e mostra. Depois, explique para que serve o ROWTYPE.

```
DECLARE
```

```
Depto_reg departamento%ROWTYPE;
```

```
BEGIN
```

```
depto_reg.coddepto := 12;
```

```
depto_reg.nome := 'Financeiro';
```

```
depto_reg.local := 'Sede';
```

```
DBMS_OUTPUT.PUT_LINE('depto_id: ' || depto_reg.coddepto);
```

```
DBMS_OUTPUT.PUT_LINE('depto_nome: ' || depto_reg.nome);
```

```
DBMS_OUTPUT.PUT_LINE('depto_local: ' || depto_reg.local);
```

```
END;
```

3. Veja a estrutura da tabela ARTISTA e seu conteúdo. Caso não exista o campo “indicacaooscar” (tipo number), adicione-o à tabela. Em seguida, verifique o seguinte código.

O que ele faz? Onde você viu o resultado?

```

BEGIN
  UPDATE artista
    SET indicacaooscar = 10
    WHERE cod_artista = 100;
  IF SQL%NOTFOUND THEN
    INSERT INTO artista (cod_artista, nome_artista, indicacaooscar)
      VALUES (100, 'CICCO',
    END IF;
END;

```

4. Faça um **trigger** na tabela *testa_bloco* (verifique se ela está criada). Em seguida, faça uma rotina semelhante à *testa_ins* (do material 11-PL/SQL), para inserir registros nessa tabela, sempre que o código gerado for múltiplo de 3 (** use a função `mod(i,j)` -> retorna o resto da divisão de i por j. Ex: `mod(i,3)`).

Verifique os registros inseridos. Como você verificou isso?

5. Verifique o seguinte código:

```

Declare Numero CONSTANT NUMBER:= 4;
BEGIN
  FOR I IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE(Numero || 'x' || I || '=' || Numero*I);
  END LOOP;
END;

```

O que ele faz?? O que ele mostra?? Altere a constante para o valor 5 e reexecute-o.

6. Agora, crie um bloco anônimo para mostrar a tabuada de números de 1 a 9, com base no código anterior. Veja como deve ser o resultado:

```

1x1 = 1
1x2 = 2
1x3 = 3
...
9x1 = 9
...
9x9 = 81

```

7. Insira mais registros na tabela **Filmes** (pelo menos 4). Implemente uma rotina para mostrar **os filmes** que pertencem à categoria com descrição = '**Ação**'. Use um CURSOR para isso. Use a cláusula FOR. Repita a rotina, dessa vez, verificando filmes de outra categoria, por exemplo, 'Ação'.

8. Verifique o seguinte bloco:

```
Declare
    cursor v_cursor_TB is select coluna1,coluna2 from testa_bloco;
Begin
    for x in v_cursor_TB loop
        dbms_output.put_line(x.coluna1 || '->' || TO_CHAR(x.coluna2,'MM/DD/YYYY,HH:MI:SS'));
    end loop;
End;
```

O que ele faz?

Reescreva-o definindo o cursor **usando FOR**.