

Goiânia, 11 de outubro de 2019
Aluna: Mylena Mariana Pires Barbosa 201804692
Prof.: Edmundo Spoto
Disciplina: Tópicos Especiais em Banco de Dados

```
SQL> SET SQLPROMPT "MYLENA:>"
```

```
MYLENA:>SET LINESIZE 500;
```

```
MYLENA:>SET PAGESIZE 500;
```

```
////////////////////////////////////  
@L03_27
```

```
MYLENA:>VARIABLE MSG VARCHAR2(100)
```

```
MYLENA:>ED
```

```
Gravou file afiedt.buf
```

```
1 DECLARE  
2     VALOR          NUMBER := &VAL;  
3 BEGIN  
4     IF VALOR > 0 THEN  
5         :MSG := 'Valor maior que zero';  
6     ELSIF VALOR = 0 THEN  
7         :MSG := 'Valor igual a zero';  
8     ELSE  
9         :MSG := 'Valor menor que zero';  
10    END IF;  
11* END;
```

```
MYLENA:>/
```

```
Informe o valor para val: 10
```

```
antigo  2:  VALOR          NUMBER := &VAL;
```

```
novo    2:  VALOR          NUMBER := 10;
```

```
Procedimento PL/SQL concluído com sucesso.
```

```
MYLENA:>PRINT MSG
```

```
MSG
```

```
-----  
Valor maior que zero
```

```
MYLENA:>/
```

```
Informe o valor para val: 0
```

```
antigo  2:  VALOR          NUMBER := &VAL;
```

```
novo    2:  VALOR          NUMBER := 0;
```

```
Procedimento PL/SQL concluído com sucesso.
```

```
MYLENA:>PRINT MSG
```

```
MSG
```

```
-----  
Valor igual a zero
```

```
MYLENA:>/
Informe o valor para val: -10
antigo 2: VALOR          NUMBER := &VAL;
novo 2: VALOR          NUMBER := -10;
```

Procedimento PL/SQL concluído com sucesso.

```
MYLENA:>PRINT MSG
```

MSG

Valor menor que zero

////////////////////////////////////

```
MYLENA:>VARIABLE MSG VARCHAR2(100)
```

```
15:14:15 MYLENA:>ED
```

Gravou file afiedt.buf

```
1 DECLARE
2     NOME EMP.ENAME%TYPE;
3     CODIGO EMP.EMPNO%TYPE:=&COD;
4     SALARIO EMP.SAL%TYPE;
5     DEPARTAMENTO DEPT.DNAME%TYPE;
6 BEGIN
7     SELECT E.ENAME, E.SAL, D.DNAME
8     INTO NOME, SALARIO, DEPARTAMENTO
9     FROM EMP E, DEPT D
10    WHERE E.EMPNO = CODIGO AND D.DEPTNO = E.DEPTNO;
11    :MSG:= DEPARTAMENTO||'-'||'EMP:'||'-'||NOME||'-'||'GANHA:'||SALARIO;
12* END;
```

```
15:18:25 MYLENA:>/
```

Informe o valor para cod: 7698

```
antigo 3: CODIGO EMP.EMPNO%TYPE:=&COD;
novo 3: CODIGO EMP.EMPNO%TYPE:=7698;
```

Procedimento PL/SQL concluído com sucesso.

```
15:18:35 MYLENA:>PRINT MSG
```

MSG

SALES-EMP:-BLAKE-GANHA:3135

```
15:18:38 MYLENA:>/
```

Informe o valor para cod: 7900

```
antigo 3: CODIGO EMP.EMPNO%TYPE:=&COD;
novo 3: CODIGO EMP.EMPNO%TYPE:=7900;
```

Procedimento PL/SQL concluído com sucesso.

```
15:18:44 MYLENA:>PRINT MSG
```

MSG

SALES-EMP:-JAMES-GANHA:1045

////////////////////////////////////
@L03_28

15:32:53 MYLENA:>ED
Gravou file afiedt.buf

```
1 DECLARE
2     SALARIO      NUMBER;
3     ENDROW       ROWID;
4 BEGIN
5     SELECT VL_SAL, ROWID INTO SALARIO, ENDROW
6     FROM FUNC
7     WHERE CD_MAT = 150;
8     IF SALARIO < 2000 THEN
9         SALARIO := SALARIO * 1.3;
10    ELSIF SALARIO IS NULL THEN
11        SALARIO := 1500;
12    ELSE
13        SALARIO := SALARIO * 1.15;
14    END IF;
15    UPDATE FUNC SET VL_SAL = SALARIO
16    WHERE ROWID = ENDROW;
17    :MSG := 'Salario = '||SALARIO;
18    COMMIT;
19* END;
15:33:13 MYLENA:>/
```

Procedimento PL/SQL concluÝdo com sucesso.

15:33:14 MYLENA:>PRINT MSG

MSG

Salario = 3343,28

////////////////////////////////////
@L03_29

15:41:31 MYLENA:>ED
Gravou file afiedt.buf

```
1 DECLARE
2     CONTA        NUMBER := 0;
3 BEGIN
4     :MSG:=' ';
5     <<INICIO>>
6     CONTA := CONTA + 1;
```

```

7      IF CONTA < 5 THEN
8          :MSG:=:MSG||' PASSEI AQUI-'||CONTA;
9          GOTO INICIO;
10     END IF;
11* END;
15:42:16 MYLENA:>/

```

Procedimento PL/SQL concluído com sucesso.

```
15:42:16 MYLENA:>PRINT MSG
```

MSG

```

-----
PASSEI AQUI-1 PASSEI AQUI-2 PASSEI AQUI-3 PASSEI AQUI-4

////////////////////////////////////
@L03_30

```

```

16:36:17 MYLENA:>ED
Gravou file afiedt.buf

```

```

1  DECLARE
2      PROCEDURE DESVIO IS
3      BEGIN
4          GOTO PARA_FORA;
5      END;
6  BEGIN
7      <<INICIO>>
8      IF TRUE THEN
9          GOTO PROXIMO_IF;
10     END IF;
11     IF FALSE THEN
12         <<PROXIMO_IF>>
13         GOTO SUBBLOCO;
14     END IF;
15     <<PARA_FORA>>
16     BEGIN
17         <<SUBBLOCO>>
18         NULL;
19     END;
20 EXCEPTION
21     WHEN OTHERS THEN
22         GOTO INICIO;
23* END;
16:36:45 MYLENA:>/
      GOTO PROXIMO_IF;
      *

```

ERRO na linha 9:
ORA-06550: linha 9, coluna 7:
PLS-00375: instrução GOTO inválida; este GOTO não pode ser ramificado para o
label 'PROXIMO_IF'
ORA-06550: linha 11, coluna 13:
PL/SQL: Statement ignored
ORA-06550: linha 4, coluna 5:

PLS-00375: instrução GOTO inválida; este GOTO não pode ser ramificado para o label 'PARA_FORA'

ORA-06550: linha 6, coluna 1:

PL/SQL: Statement ignored

ORA-06550: linha 22, coluna 5:

PLS-00375: instrução GOTO inválida; este GOTO não pode ser ramificado para o label 'INICIO'

ORA-06550: linha 22, coluna 5:

PL/SQL: Statement ignored

```
////////////////////////////////////  
/////////  
@L03_31
```

16:38:53 MYLENA:>ED

Gravou file afiedt.buf

```
1 DECLARE  
2   VEZ          NUMBER := 1;  
3 BEGIN  
4   <<INICIO>>  
5   VEZ := VEZ + 1;  
6   IF VEZ > 10 THEN  
7     GOTO FIM;  
8   ELSE  
9     VEZ := VEZ**2 - 1;  
10  END IF;  
11  GOTO INICIO;  
12  <<FIM>>  
13  --NULL;  
14* END;
```

16:39:12 MYLENA:>/

END;

*

ERRO na linha 14:

ORA-06550: linha 14, coluna 1:

PLS-00103: Encontrado o símbolo "END" quando um dos seguintes símbolos era esperado:

(começar case declare sair for goto if loop mod nulo raise
retornar selecionar atualizar while com <um identificador>
<um identificador delimitado por aspas duplas>
<uma variável de ligação> << continuar fechar atual
excluir fetch lock insert aberto rollback savepoint set sql
executar commit forall intercalar pipe expurgar json_exists
json_value json_query json_object json_array

16:39:13 MYLENA:>ED

Gravou file afiedt.buf

```
1 DECLARE  
2   VEZ          NUMBER := 1;  
3 BEGIN  
4   <<INICIO>>  
5   VEZ := VEZ + 1;
```

```

6      IF VEZ > 10 THEN
7          GOTO FIM;
8      ELSE
9          VEZ := VEZ**2 - 1;
10     END IF;
11     GOTO INICIO;
12 <<FIM>>
13     NULL;
14* END;
16:39:57 MYLENA:>/

```

Procedimento PL/SQL concluído com sucesso.

////////////////////////////////////

```

17:03:42 MYLENA:>ed
Gravou file afiedt.buf

```

```

1  declare
2      v_nome varchar2(30);
3      v_sal number(8,2);
4  begin
5      select NM_FUNC, VL_SAL
6          INTO v_nome, v_sal
7          From func
8          where CD_MAT=&cod;
9      dbms_output.put_line(v_nome||'  '||v_sal);
10     exception
11         when no_data_found then
12             dbms_output.put_line('Funcionario nao Cadastrado');
13* end;

```

```

17:03:59 MYLENA:>/
Informe o valor para cod: 330
antigo   8:      where CD_MAT=&cod;
novo    8:      where CD_MAT=330;
WILSON  2537

```

Procedimento PL/SQL concluído com sucesso.

```

17:04:22 MYLENA:>/
Informe o valor para cod: 120
antigo   8:      where CD_MAT=&cod;
novo    8:      where CD_MAT=120;
SILVIO  2925

```

Procedimento PL/SQL concluído com sucesso.

```

17:04:31 MYLENA:>/
Informe o valor para cod: 122
antigo   8:      where CD_MAT=&cod;
novo    8:      where CD_MAT=122;
Funcionario nao Cadastrado

```

Procedimento PL/SQL concluído com sucesso.

//

17:04:35 MYLENA:>ed

Gravou file afiedt.buf

```
1 declare
2   v_nome  FUNC.NM_FUNC%TYPE;
3   v_sal    FUNC.VL_SAL%TYPE;
4 begin
5   select NM_FUNC, VL_SAL
6     INTO v_nome, v_sal
7     From   FUNC;
8   dbms_output.put_line(v_nome||' '||v_sal);
9   exception
10    when no_data_found then
11      dbms_output.put_line('Funcionario nao Cadastrado');
12    when too_many_rows then
13      dbms_output.put_line('Retornou muitos registros');
14* end;
```

17:08:05 MYLENA:>/

Retornou muitos registros

Procedimento PL/SQL concluído com sucesso.

17:10:33 MYLENA:>ed

Gravou file afiedt.buf

```
1 declare
2   v_nome  FUNC.NM_FUNC%TYPE;
3   v_sal    FUNC.VL_SAL%TYPE;
4 begin
5   select NM_FUNC, VL_SAL
6     INTO v_nome, v_sal
7     From   FUNC
8     WHERE cd_mat = &cod;
9   dbms_output.put_line(v_nome||' '||v_sal);
10  exception
11   when no_data_found then
12     dbms_output.put_line('Funcionario nao Cadastrado');
13   when too_many_rows then
14     dbms_output.put_line('Retornou muitos registros');
15* end;
```

17:11:04 MYLENA:>/

Informe o valor para cod: 120

antigo 8: WHERE cd_mat = &cod;

novo 8: WHERE cd_mat = 120;

SILVIO 2925

Procedimento PL/SQL concluído com sucesso.

17:11:07 MYLENA:>/

Informe o valor para cod: 122

```
antigo 8:      WHERE cd_mat = &cod;
novo 8:      WHERE cd_mat = 122;
Funcionario nao Cadastrado
```

Procedimento PL/SQL concluído com sucesso.

```
////////////////////////////////////
IF - %FOND
```

```
17:11:12 MYLENA:>ed
Gravou file afiedt.buf
```

```
1  begin
2    update FUNC set
3      VL_SAL=240
4    where VL_SAL<200;
5    if SQL%FOUND then
6      commit;
7    else
8      dbms_output.put_line('Valores nao Encontrados');
9    end if;
10* end;
```

```
17:20:52 MYLENA:>/
Valores nao Encontrados
```

Procedimento PL/SQL concluído com sucesso.

```
////////////////////////////////////
@L03_82
```

```
17:28:35 MYLENA:>SET LINESIZE 70
17:28:50 MYLENA:>ED
Gravou file afiedt.buf
```

```
1  DECLARE
2    TYPE TREG IS RECORD (NOME      VARCHAR2(30),
3                          MAT       NUMBER NOT NULL := 0,
4                          DEP       DEPTO.CD_DEPTO%TYPE);
5    VREG      TREG;
6    CURSOR C1 IS SELECT NM_FUNC||' '||NM_SOBRENOME, CD_MAT,
7                      CD_DEPTO FROM FUNC;
8  BEGIN
9    OPEN C1;
10   :MSG := '';
11   LOOP
12     FETCH C1 INTO VREG;
13     EXIT WHEN C1%NOTFOUND;
14     IF VREG.MAT > 200 AND VREG.DEP IN ('D11', 'E11') THEN
15       :MSG := :MSG || VREG.NOME ||'; ';
16     END IF;
17   END LOOP;
18* END;
17:28:56 MYLENA:>/
```


Procedimento PL/SQL concluído com sucesso.

17:28:57 MYLENA:>PRINT MSG

MSG

WILLIAM JONES; JOANA LUZ; ELINE SEVERO; JOAO PONTES; FELIPE SARAIVA; MARINA SALGADO;

17:34:38 MYLENA:>ED

Gravou file afiedt.buf

```
1 DECLARE
2     TYPE TREG IS RECORD (NOME      VARCHAR2(30),
3                           MAT       NUMBER NOT NULL := 0,
4                           DEP       DEPTO.CD_DEPTO%TYPE);
5     VREG      TREG;
6     CURSOR C1 IS SELECT NM_FUNC||' '||NM_SOBRENOME, CD_MAT,
7                        CD_DEPTO FROM FUNC;
8 BEGIN
9     OPEN C1;
10    LOOP
11        FETCH C1 INTO VREG;
12        EXIT WHEN C1%NOTFOUND;
13        IF VREG.MAT > 200 AND VREG.DEP IN ('D11', 'E11') THEN
14            DBMS_OUTPUT.PUT_LINE(VREG.NOME||' - '||VREG.MAT||' - DEPARTAMENTO:
15            '||VREG.DEP);
16        END IF;
17    END LOOP;
18* END;
```

17:34:53 MYLENA:>/

WILLIAM JONES - 210 - DEPARTAMENTO: D11
JOANA LUZ - 220 - DEPARTAMENTO: D11
ELINE SEVERO - 280 - DEPARTAMENTO: E11
JOAO PONTES - 290 - DEPARTAMENTO: E11
FELIPE SARAIVA - 300 - DEPARTAMENTO: E11
MARINA SALGADO - 310 - DEPARTAMENTO: E11

Procedimento PL/SQL concluído com sucesso.