

FIAP GRADUAÇÃO

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Database Application Development

PROF. LEONARDO UCHIDA

CONTEÚDO DO CURSO

Variáveis

- O que são variáveis em PL/SQL
- Como definir variáveis / constantes
- Exemplos

Operadores

- Operadores Aritméticos
- Operadores Relacionais
- Operadores de Comparação
- Operadores Lógicos
- Ordem de Precedência

Estruturas de Controle

- IF
- CASE
- ITERADOR
 - LOOP
 - WHILE
 - FOR

Variáveis

O que são Espaços Reservados em PL/SQL?

- São espaços reservados para armazenamento temporário de dados, que residirão na memória RAM.
- Podem ser definidos como: variáveis, constantes(não modificável em tempo de execução) e registros.
- São utilizadas somente em tempo de execução do bloco PL/SQL.

Declaração de variáveis em PL/SQL

- As variáveis devem ser declaradas e inicializadas na seção de declaração, ou em packages como variáveis globais.
- Atribuir novos valores às variáveis na seção executável, exceto para constantes.
- Variáveis com precisão definidas ocupam menos espaço de memória.
- Cada Espaço Reservado deve ter um tipo de dado específico.

Declaração de variáveis em PL/SQL

- Não são case-sensitive.
- Nomes devem ser únicos, mesmo que os datatypes sejam diferentes.
- Caso o nome da variável seja igual ao nome de uma coluna de uma tabela, o nome da coluna sempre é prioritário em relação à variável.

Sintaxe:

```
variable_name [CONSTANT] datatype [NOT NULL] [:= | DEFAULT  
initial_value]
```

Exemplos para declaração de variáveis

--Declaração com / sem inicialização

```
DECLARE
  birthday DATE;
  emp_count SMALLINT := 0;
BEGIN
  NULL;
END;
```

--Declaração complexa, com dependência.

```
DECLARE
  pi REAL := 3.14159;
  radius REAL := 1;
  area REAL := pi * radius**2;
BEGIN
  NULL;
END;
```


Exemplos para declaração de variáveis

--Declaração de constantes

```
DECLARE
  credit_limit CONSTANT REAL := 5000.00;
  max_days    CONSTANT INTEGER := 366;
  paid        CONSTANT BOOLEAN := FALSE;
BEGIN
  NULL;
END;
```

--Declaração utilizando DEFAULT.

```
DECLARE
  hours_worked  INTEGER DEFAULT 40;
  employee_count INTEGER := 0;
BEGIN
  NULL;
END;
```

Exemplos para declaração de variáveis

--Declaração restrição NOT NULL

```
DECLARE
  acct_id INTEGER(4) NOT NULL := 9999;
BEGIN
  NULL;
END;
```

--Declaração tipo BOOLEAN.

```
DECLARE
  done BOOLEAN; --inicialmente é NULO
BEGIN
  done := FALSE; -- Atribuição de valor
  done := TRUE; -- Reatribuição de valor
END;
```

Exemplo de atribuição de valores

--Atribuição com base em coluna do banco de dados

DECLARE

v_nome_depto varchar2(20);

v_id_depto number := 10;

BEGIN

 select cd_depto, nm_depto

 into v_id_depto, v_nome_depto

 from loc_depto

 where cd_depto = v_cd_depto;

 dbms_output.put_line(v_nome_depto);

END;

Datatype %TYPE

- O %TYPE permite a declaração de variáveis com o mesmo tipo de dado de outra variável ou coluna de tabela, mesmo não sabendo o tipo.
- Caso o item de referência master tenha mudado, automaticamente suas referencias também se modificam.
- É também conhecido como herança de tipo de dado de coluna, mas não herda valores.

Exemplo:

```
DECLARE
  nomecompleto    loc_funcionario.nm_func%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE('nome=' || nomecompleto);
END;
```

Datatype %ROWTYPE

- O %ROWTYPE permite a declaração de variáveis que permitem armazenar um registro completo de uma tabela ou view.
- O registro possui o mesmo nome dos campos e tipo de dado.
- Os campos da variável %ROWTYPE não herdam as constraints ou valores DEFAULT.
- Se a tabela de origem for modificada, a declaração automaticamente se adapta.
- Para referenciar um campo da variável, utilizar VARIÁVEL.CAMPO.

Datatype %ROWTYPE

Exemplo:

```
DECLARE
  v_reg_func  loc_funcionario%ROWTYPE;
BEGIN
  select *
    into  v_reg_func
    from  loc_funcionario
   where cd_func=1;
  ...
  dbms_output.put_line(v_reg_func.nm_func);
END;
```

Operadores

OPERADORES ARITMÉTICOS

EXEMPLO

```
A  NUMBER := 10;  
B  NUMBER := 5;
```

OPERADOR	DESCRIÇÃO	EXEMPLO
+	Adição	$A + B = 15$
-	Subtração	$A - B = 5$
*	Multiplicação	$A * B = 50$
/	Divisão	$A / B = 2$
**	Exponenciação	$A ** B = 100.000$

Exemplo:

```
BEGIN  
  dbms_output.put_line( 10 + 5 );  
  dbms_output.put_line( 10 - 5 );  
  dbms_output.put_line( 10 * 5 );  
  dbms_output.put_line( 10 / 5 );  
  dbms_output.put_line( 10 ** 5 );  
END;  
/
```

Resultado:

```
15  
5  
50  
2  
100000
```

PL/SQL procedure successfully completed.

OPERADORES RELACIONAIS

EXEMPLO

A NUMBER := 10;

B NUMBER := 5;

OPERADOR	DESCRIÇÃO	EXEMPLO
=	Igualdade	(A=B) Falso
!= <> ~=	Diferente	(A!=B) Verdadeiro
>	Maior	(A>B) Verdadeiro
<	Menor	(A<B) Falso
>= <=	Maior igual ou Menor igual	(A>=B) Verdadeiro (A<=B) Falso

OPERADORES DE COMPARAÇÃO

SEMPRE RETORNA

VERDADEIRO

FALSO

NULO

OPERADOR	DESCRIÇÃO	EXEMPLO
LIKE	Retorna verdadeiro se o padrão especificado for atendido	Se 'ABC' LIKE 'A%' retorna verdadeiro
BETWEEN	Retorna verdadeiro se o valor estiver no intervalo especificado	Se x=5 então, x between 1 e 10 é verdadeiro
IN	Retorna verdadeiro para todas as ocorrências encontradas no conjunto	Se x='A' então, x in ('X','B','A') retorna verdadeiro
IS NULL	Testa ausência de valor	Se x = null então, x is null é verdadeiro

| OPERADORES LÓGICOS

EXEMPLO

A BOOLEAN := TRUE;
B BOOLEAN := FALSE;

OPERADOR	DESCRIÇÃO	EXEMPLO
AND	Testa ambos os casos	(A and B) é falso
OR	Testa somente um dos casos	(A or B) é verdadeiro
NOT	Lógica reversa para AND ou OR	NOT (A and B) é verdadeiro

PRECEDÊNCIA DOS OPERADORES

Ordem lógica e matemática.

Exemplo:

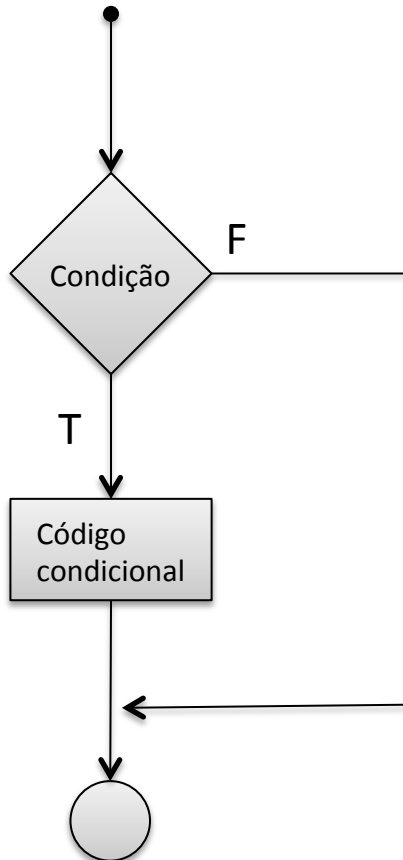
$x=2+3*5$ (multiplicação e divisão tem prioridade)
 $x=17$

Ordem para relacionais e de comparação:

$=$, $<$, $>$, $<=$, $>=$, $<>$, $!=$, $\sim=$, $\hat{=}$, IS NULL, LIKE, BETWEEN, IN.

Estruturas de Controle

ESTRUTURAS DE CONTROLE (IF)



Exemplo:

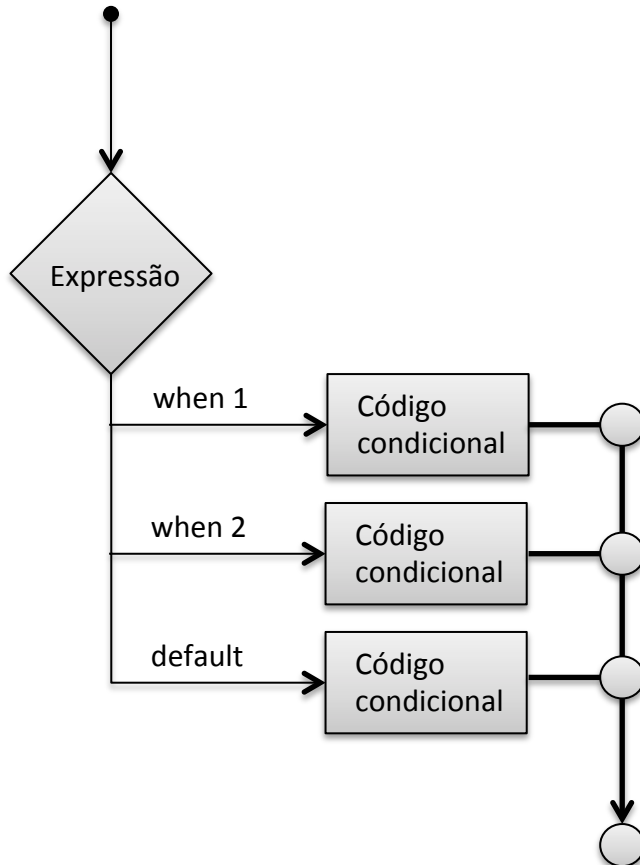
```
DECLARE
  a number(3) := 100;
BEGIN
  IF ( a = 10 ) THEN
    dbms_output.put_line('Valor de a é 10' );
  ELSIF ( a = 20 ) THEN
    dbms_output.put_line('Valor de a é 20' );
  ELSIF ( a = 30 ) THEN
    dbms_output.put_line('Valor de a é 30' );
  ELSE
    dbms_output.put_line('Valor não encontrado');
  END IF;
  dbms_output.put_line('O valor exato de a é: ' || a );
END;
```

Resultado:

Valor não encontrado
O Valor exato de a é: 100

PL/SQL procedure successfully completed.

ESTRUTURAS DE CONTROLE (CASE)



Exemplo:

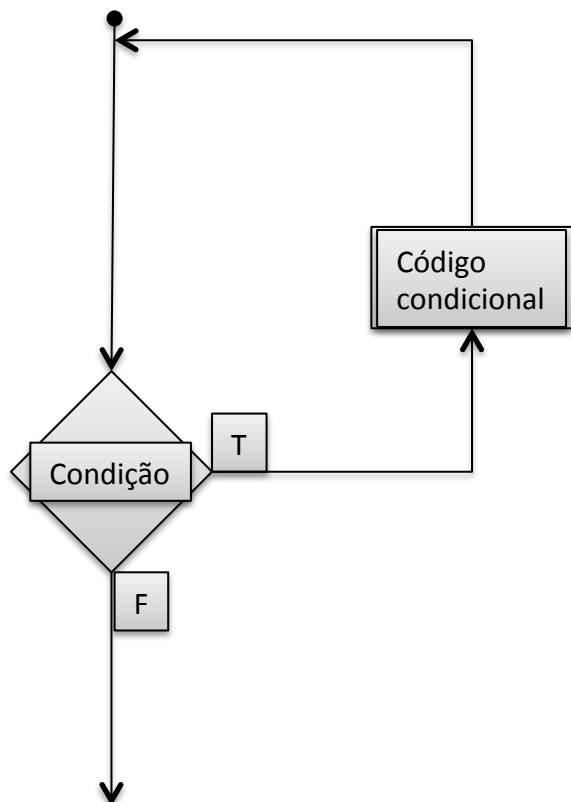
```
DECLARE
  grade char(1) := 'A';
BEGIN
  CASE grade
    when 'A' then dbms_output.put_line('Excelente');
    when 'B' then dbms_output.put_line('Parabéns');
    when 'C' then dbms_output.put_line('Muito bom');
    when 'D' then dbms_output.put_line('Aprovado');
    when 'F' then dbms_output.put_line('Tente novamente');
    else dbms_output.put_line('Sem nota');
  END CASE;
END;
```

Resultado:

Excelente

PL/SQL procedure successfully completed.

ITERAÇÃO (LOOP)



Exemplo:

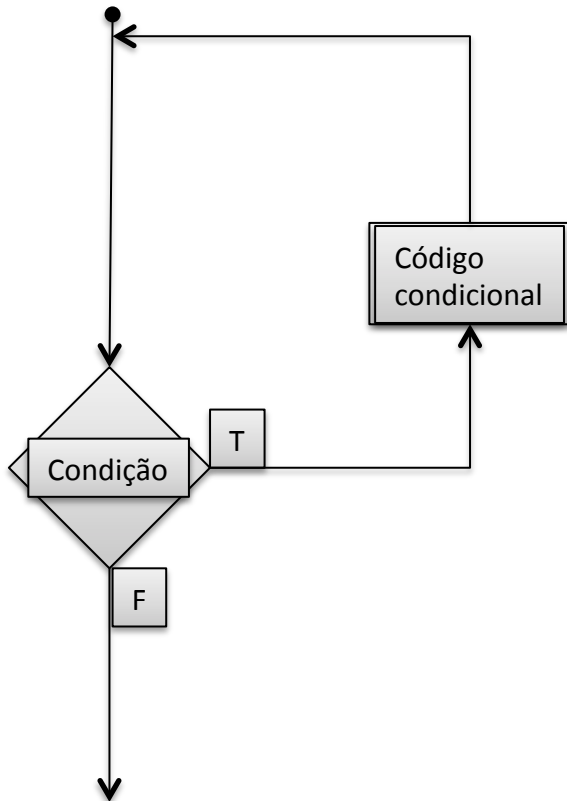
```
DECLARE
  x number := 10;
BEGIN
  LOOP
    dbms_output.put_line(x);
    x := x + 10;
    IF x > 50 THEN
      exit;
    END IF;
  END LOOP;
  dbms_output.put_line('Valor de x é: ' || x);
END;
```

Resultado:

```
10
20
30
40
50
Valor de x é: 60
```

PL/SQL procedure successfully completed.

ITERAÇÃO (WHILE)



Exemplo:

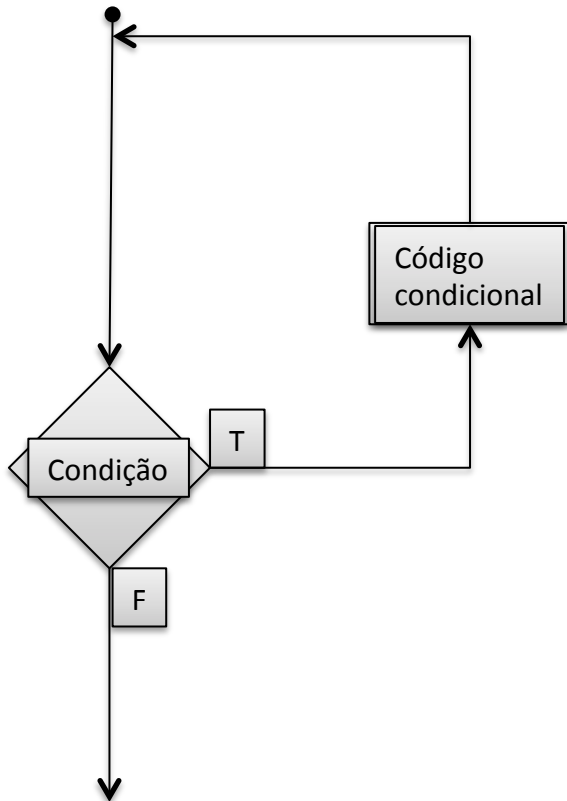
```
DECLARE
  a number(2) := 10;
BEGIN
  WHILE a < 20 LOOP
    dbms_output.put_line('value of a: ' || a);
    a := a + 1;
  END LOOP;
END;
```

Resultado:

```
valor de a: 10
valor de a: 11
valor de a: 12
valor de a: 13
valor de a: 14
valor de a: 15
valor de a: 16
valor de a: 17
valor de a: 18
valor de a: 19
```

PL/SQL procedure successfully completed.

ITERAÇÃO (FOR)



Exemplo:

```
DECLARE
  a number(2);
BEGIN
  FOR a in 10 .. 20 LOOP
    dbms_output.put_line('valor de a: ' || a);
  END LOOP;
END;
```

Resultado:

```
valor de a: 10
valor de a: 11
valor de a: 12
valor de a: 13
valor de a: 14
valor de a: 15
valor de a: 16
valor de a: 17
valor de a: 18
valor de a: 19
valor de a: 20
```

PL/SQL procedure successfully completed.

Vamos revisar nosso aprendizado??

<https://bit.ly/2GJmZgL>

I REFERÊNCIAS



- Manuais ou Materiais oficiais da Oracle ou Oracle Press
- Oracle9i / 10g PL/SQL * Guia de Consulta Rápida – Celso Henrique Poderoso

Copyright © 2019 Prof. Leonardo Uchida

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).