

Arquitetura e Administração de Bases de Dados
Oracle11g PL/SQL Programming

João Costa
jcosta@isec.pt

1

Agenda

Procedimentos

- ▶ **CREATE PROCEDURE ...**
- ▶ **DROP PROCEDURE ...**



2023/24 - LEI - AABD - PL/SQL

2

Procedimentos

- ▶ Um procedimento é um bloco PL/SQL com um nome e que executa uma dada ação
- ▶ Pode ser armazenado na base de dados como um objeto para execuções repetidas.



2023/24 - LEI - AABD - PL/SQL

3

Procedimentos

Header

```
CREATE [OR REPLACE] PROCEDURE nome_procedimento
[ (parametro1 [modo] datatype,
  parametro2 [modo] datatype, ... ) ]
IS | AS
```

Bloco PL/SQL

```
-- declaração de variáveis
BEGIN
-- Código

EXCEPTION
-- tratamento de exceções
END;
```



2023/24 - LEI - AABD - PL/SQL

4

Parâmetros

- ▶ Tornar os programas reutilizáveis
- ▶ Usados para enviar dados **de** e **para** os programas
 - ▶ Modos :
 - ▶ IN
 - ▶ OUT
 - ▶ IN OUT



2023/24 - LEI - AABD - PL/SQL

5

Parâmetros (cont.)

Modo IN

- ▶ Modo por defeito, se nenhum for especificado
- ▶ Passa um valor do ambiente da aplicação para o procedimento
- ▶ O valor é considerado constante
 - ▶ Não pode ser alterado dentro do procedimento



2023/24 - LEI - AABD - PL/SQL

6

Parâmetros (cont.)

Modo OUT

- ▶ Permite devolver valores para o programa que chamou o subprograma.
- ▶ Dentro do procedimento funciona como uma variável local
- ▶ Passa um valor do procedimento para o ambiente da aplicação
- ▶ Valores calculados ou obtidos da BD
 - ▶ Podem ser devolvidos usando parâmetros OUT



2023/24 - LEI - AABD - PL/SQL

7

Parâmetros (cont.)

Modo IN OUT

- ▶ Permite usar o mesmo parâmetro para
 - ▶ Passar um valor para dentro do procedimento
 - ▶ Passar um valor para fora do procedimento
 - ▶ Passar os valores iniciais da variável para o procedimento e devolver esses valores atualizados para o programa que o chamou
 - O valor no programa só é atualizado depois do procedimento terminar
- ▶ Os valores passados ao procedimento podem ser diferentes dos devolvidos
- ▶ Dentro do procedimento funciona como uma variável local.



2023/24 - LEI - AABD - PL/SQL

8

Criar procedimentos: Exemplo

► Determinar o total a pagar.

```
CREATE OR REPLACE PROCEDURE calcula_total (
  QUANT IN NUMBER,
  PRECO IN NUMBER,
  TOTAL OUT NUMBER)
IS
BEGIN
  TOTAL := QUANT * PRECO;
  IF QUANT > 10 THEN
    TOTAL := TOTAL * 0.95;
  END IF;
END;
/

Procedure created
```

modo IN

modo IN

modo OUT

2023/24 - LEI - AABD - PL/SQL

9

Executar um procedimento

```
VARIABLE tot_cost NUMBER
CALL calcula_total(12,10, :tot_cost );
Procedure successfully completed.

PRINT tot_cost

TOT_COST
-----
114
```

Declarar uma variável host

Usar o comando
CALL

Passar a variável host

Usar o comando
PRINT do SQL*Plus
para visualizar
o conteúdo da variável.

Nota: Os argumentos são passados de forma posicional

2023/24 - LEI - AABD - PL/SQL

10

Executar um procedimento

```

VARIABLE tot_cost NUMBER
EXECUTE calcula_total(12,10, :tot_cost );
Procedure successfully completed.

PRINT tot_cost
TOT_COST
-----
    114

```

Declaração de uma variável host

Usar o comando **EXECUTE** do SQL*Plus

Passar a variável host

Usar o comando **PRINT** do SQL*Plus para visualizar o conteúdo da variável.

Nota: Os argumentos são passados de forma posicional

2023/24 - LEI - AABD - PL/SQL

11

O comando SHOW ERRORS

```

CREATE OR REPLACE PROCEDURE calcula_total
(QUANT IN NUMBER, PRECO IN NUMBER(3), TOTAL OUT NUMBER)
IS
BEGIN
    TOTAL := QUANT * PRECO;
    IF QUANT > 10 THEN
        TOTAL := TOTAL * 0.95;
    END IF;
END;
/
Warning: Procedure created with compilation errors.

SHOW ERRORS
Errors for PROCEDURE calcula_total
LINE/COL ERROR
-----
02/23      PLS-00103: encountered the symbol "(" when
           expecting one of the following:
           := . ) , @ default character
           The symbol ":" was substituted for "(" to continue;

```

Erro ao indicar o tamanho do parâmetro

Erro de compilação

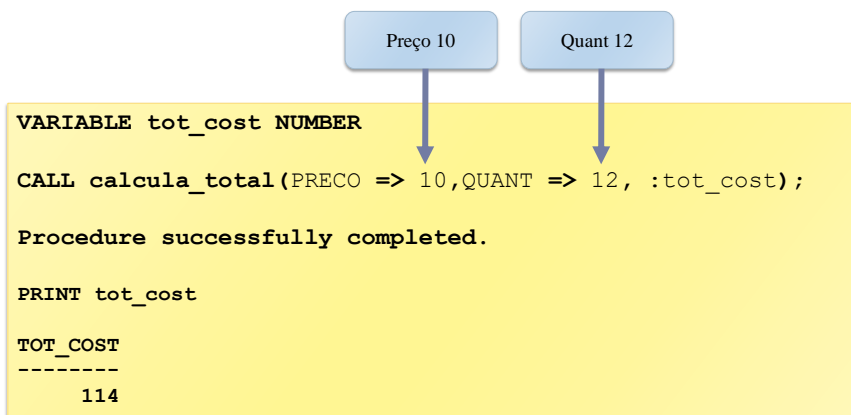
Mostrar a mensagem de erro

2023/24 - LEI - AABD - PL/SQL

12

Passar parâmetros pelo nome

- ▶ Usar “=>” e o nome do parâmetro



2023/24 - LEI - AABD - PL/SQL

13

Parâmetros com valores por defeito

```

CREATE PROCEDURE TEST(
    a number,
    b number := 10,
    d date default sysdate )
IS
BEGIN
    null;
END;
  
```

Se não definido, assume 10

Se não definido,
assume o instante atual

Assume

CALL test(12);

(12, 10, hoje)

CALL test(12,5);

(12, 5, hoje)

CALL test(12,5, sysdate-1);

(12, 5, ontem)

CALL test(a => 12,d => sysdate+1);

(12, 10, amanhã)

2023/24 - LEI - AABD - PL/SQL

14

Modo IN OUT

- ▶ Enviar e receber um valor através do mesmo parâmetro

```
CREATE OR REPLACE PROCEDURE phone_fmt
  (p_phone IN OUT VARCHAR2)
IS
BEGIN
  p_phone := '(' || SUBSTR(p_phone,1,5) || ') ' ||
             SUBSTR(p_phone,6,2) || ' ' ||
             SUBSTR(p_phone,8) ;
END;
/
```

00351912345678 → (00351) 91 2345678

2023/24 - LEI - AABD - PL/SQL

15

Passar os valores dos parâmetros

- ▶ Duas técnicas para passar valores entre parâmetros efetivos e formais
 - **Por valor**
copia o valor do parâmetro efetivo para o formal
 - **Por referência**
criado um ponteiro para o parâmetro efetivo
- ▶ Por defeito: passagem por valor

2023/24 - LEI - AABD - PL/SQL

16

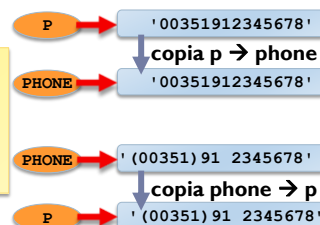
Passagem por valor

► Enviar por valor

```
DECLARE
  p VARCHAR(30) := '00351912345678';
BEGIN
  DBMS_OUTPUT.PUT_LINE('Phone = ' || p);
  phone_fmt(p);
  DBMS_OUTPUT.PUT_LINE('Phone = ' || p);
END;
```

00351912345678 → (00351) 91 2345678

```
CREATE PROCEDURE phone_fmt(phone IN OUT VARCHAR2)
IS
BEGIN
  phone := (' || SUBSTR(phone,1,5) || ' ) ' ||
    SUBSTR(phone,6,2) || ' ' || SUBSTR(phone,8);
END;
```



2023/24 - LEI - AABD - PL/SQL

17

Passagem por referência

► “Pedido” ao compilador para passar por referência

```
CREATE OR REPLACE PROCEDURE calcula_total (
  quant IN NUMBER,
  preco IN NUMBER,
  total IN OUT NOCOPY NUMBER )
IS
BEGIN
  total := quant * preco;
  IF quant > 10 THEN
    total := total * 0.95;
  END IF;
END;
```

Compiler Hint

2023/24 - LEI - AABD - PL/SQL

18

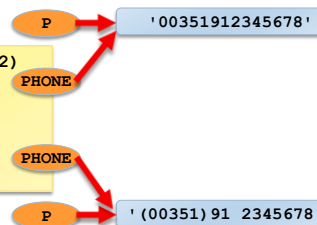
Passagem por referência

► Enviar por valor

```
DECLARE
  p VARCHAR(30) := '00351912345678';
BEGIN
  DBMS_OUTPUT.PUT_LINE('Phone = ' || p);
  phone_fmt(p);
  DBMS_OUTPUT.PUT_LINE('Phone = ' || p);
END;
```

00351912345678 → (00351) 91 2345678

```
CREATE PROCEDURE phone_fmt(phone IN OUT NOCOPY VARCHAR2)
IS
BEGIN
  phone := '(' || SUBSTR(phone, 1, 5) || ') ' ||
    SUBSTR(phone, 6, 2) || ' ' || SUBSTR(phone, 8);
END;
```



2023/24 - LEI - AABD - PL/SQL

19

Chamar um procedimento num bloco

```
SQL> CREATE OR REPLACE PROCEDURE order_total_sp
2   (p_bsktid IN NUMBER,
3   p_cnt OUT NUMBER,
4   p_sub OUT NUMBER,
5   p_ship OUT NUMBER,
6   p_total OUT NUMBER)
7   IS
8   BEGIN
9   SELECT SUM(quantity), SUM(quantity*price)
10  INTO p_cnt, p_sub
11  FROM bb_basketitem
12  WHERE idbasket = p_bsktid;
13  ship_cost_sp(p_cnt, p_ship);
14  p_total := NUL(p_sub, 0) + NUL(p_ship, 0);
15  END;
16  /

Procedure created.

SQL> |
```

2023/24 - LEI - AABD - PL/SQL

20

Comando DESCRIBE

- ▶ Permite listar os parâmetros de um programa

```
DESCRIBE calcula_total
```

```
PROCEDURE calcula_total
Argument Name  Type      In/Out  Default?
-----
QUANT          NUMBER    IN
PRECO          NUMBER    IN
TOTAL          NUMBER    IN OUT
```



2023/24 - LEI - AABD - PL/SQL

21

Comando RETURN em procedimentos

- ▶ Usado para alterar o fluxo de execução
- ▶ Termina o processamento do bloco
 - ▶ Vai para o comando seguinte à chamada do procedimento
- ▶ Usado **stand-alone** – sem argumentos



2023/24 - LEI - AABD - PL/SQL

22

Apagar um procedimento

```
DROP PROCEDURE procedure_name;
```



2023/24 - LEI - AABD - PL/SQL

23

Exercícios

- a) Criar um procedimento que recebe como argumento o código de um livro, e devolve por argumento há quantos dias foi editado.
- b) Crie um procedimento que atualize o género preferido de um autor, com o género do último livro que escreveu. Ignore os autores que não escreveram livros.



2023/24 - LEI - AABD - PL/SQL

24