

Arquitetura e Administração de Bases de Dados
Oracle10g PL/SQL Programming

João Costa
jcosta@isec.pt

Agenda

Procedimentos

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

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.

▶ 3

2021/22 - LEI - AABD - PL/SQL

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;
```

▶ 4

2021/22 - LEI - AABD - PL/SQL

Parâmetros

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

▶ 5

2021/22 - LEI - AABD - PL/SQL

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

▶ 6

2021/22 - LEI - AABD - PL/SQL

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

▶ 7

2021/22 - LEI - AABD - PL/SQL

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.

▶ 8

2021/22 - LEI - AABD - PL/SQL

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

► 9

2021/22 - LEI - AABD - PL/SQL

Executar um procedimento

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

PRINT tot_cost

TOT_COST
-----
    114
```

Declarar 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

► 10

2021/22 - LEI - AABD - PL/SQL

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;
/
```

Erro ao indicar o tamanho do parâmetro

Erro de compilação

Warning: Procedure created with compilation errors.

SHOW ERRORS

```
Errors for PROCEDURE calcula_total
LINE/COL ERROR
```

Mostrar a mensagem de erro

```
-----
02/23      PLS-00103: encountered the symbol "(" when
expecting one of the following:
:= . ) , @ default character
The symbol ":=" was substituted for "(" to continue;
```

► 11

2021/22 - LEI - AABD - PL/SQL

Passar parâmetros pelo nome

- Usar “=>” e o nome do parâmetro

Preço 10

Quant 12

```
VARIABLE tot_cost NUMBER

EXECUTE calcula_total(PRECO => 10,QUANT => 12, :tot_cost);

Procedure successfully completed.

PRINT tot_cost

TOT_COST
-----
114
```

► 12

2021/22 - LEI - AABD - PL/SQL

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

```
execute test(12);
```

(12, 10, hoje)

```
execute test(12,5);
```

(12, 5, hoje)

```
execute test(12,5, sysdate-1);
```

(12, 5, ontem)

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

(12, 10, amanhã)

► 13

2021/22 - LEI - AABD - PL/SQL

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

► 14

2021/22 - LEI - AABD - PL/SQL

Passar os valores dos parâmetros

- ▶ Duas técnicas para passar valores entre parâmetros efetivos e formais

1. Por referência

criado um ponteiro para o parâmetro efetivo

1. Por valor

copia o valor do parâmetro efetivo para o formal

- ▶ Por defeito: passagem por valor

▶ 15

2021/22 - LEI - AABD - PL/SQL

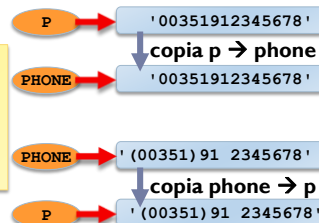
Passar os valores dos parâmetros

- ▶ 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;
```



▶ 16

2021/22 - LEI - AABD - PL/SQL

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

▶ 17

2021/22 - LEI - AABD - PL/SQL

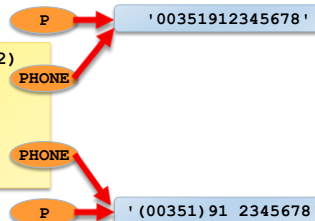
Passar os valores dos parâmetros

- ▶ 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;
```



▶ 18

2021/22 - LEI - AABD - PL/SQL

Chamar um procedimento num bloco

The screenshot shows the Oracle SQL*Plus interface. The main window displays the following SQL code:

```
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  /
```

Below the code, the message "Procedure created." is displayed. The prompt "SQL> |" is at the bottom. A callout box with an arrow pointing to line 13 contains the text "Chamada ao procedimento ship_cost_sp".

▶ 19

2021/22 - LEI - AABD - PL/SQL

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
```

▶ 20

2021/22 - LEI - AABD - PL/SQL

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

▶ 21

2021/22 - LEI - AABD - PL/SQL

Apagar um procedimento

```
DROP PROCEDURE procedure_name;
```

▶ 22

2021/22 - LEI - AABD - PL/SQL

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.