

Subprogramas

Laboratório de Bases de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Subprogramas

- Blocos PL/SQL armazenados de forma compilada, e executados no servidor
- Características de um subprograma
 - nome
 - máximo de 30 caracteres
 - modo dos parâmetros
 - **IN** (*de entrada*; passagem por valor)
 - **OUT** (*de saída*; passagem por referência)
 - **IN OUT** (*de entrada/saída*; passagem por referência)
 - não possui o comando DECLARE

Procedures

- Subprogramas que têm por objetivo executar uma ação específica
- Restrições
 - não são utilizados em atribuições
 - não podem ser usados como argumentos de um comando SELECT

Estrutura

```
CREATE OR REPLACE PROCEDURE nome_proc  
  (parâmetro1 modo tipo_de_dados,  
   parâmetro2 modo tipo_de_dados,  
   ...  
   parâmetro3 modo tipo_de_dados)  
IS ou AS  
  área de declaração  
BEGIN  
  área de comandos  
END nome_proc;
```

Exemplo

```
CREATE OR REPLACE PROCEDURE calcula
    (x IN NUMBER, y OUT NUMBER,
    z IN OUT NUMBER)
IS
BEGIN
    y := x * 2;
    y := y + z;
    z := y + x + z;
END calcula;
```

Exemplo

```
set serveroutput on;
DECLARE
    var1 NUMBER(3);
    var2 NUMBER(3);
    var3 NUMBER(3);
BEGIN
    var1 := 30;
    var2 := 40;
    var3 := 50;
    dbms_output.put_line ( var1 || ' ' || var2 || ' ' || var3 );
    calcula (var1, var2, var3);
    dbms_output.put_line ( var1 || ' ' || var2 || ' ' || var3 );
END;
```

SHOW ERRORS

- Mostra erros de compilação
 - executado após a tentativa de compilação de um procedimento
- Exemplo
 - show errors;

```
Errors for PROCEDURE CALCULA:  
LINE/COL ERROR
```

```
-----  
5/7      PLS-00363: expression 'X' cannot be used as an  
         assignment target  
5/7      PL/SQL: Statement ignored
```

Procedures

- Execução

```
EXECUTE nome_proc (lista_de_parâmetros);
```

- Remoção do banco de dados

```
DROP PROCEDURE nome_proc;
```


Functions

- Subprogramas que têm por objetivo retornar algum resultado ou valor
- Funcionalidades
 - podem ser utilizadas em atribuições
 - podem ser usadas como argumentos de um comando SELECT
 - podem ser recursivas

Estrutura

```
CREATE OR REPLACE FUNCTION nome_func  
  (parâmetro1 modo tipo_de_dados,  
   parâmetro2 modo tipo_de_dados,  
   ...  
   parâmetro3 modo tipo_de_dados)  
RETURN tipo_de_dados  
IS ou AS  
  área de declaração  
BEGIN  
  área de comandos  
END nome_func;
```

Exemplo

```
CREATE OR REPLACE FUNCTION soma_numeros
  (telefone IN VARCHAR2)
RETURN NUMBER
IS
  s NUMBER(4) := 0;
  soma NUMBER(4) := 0;
BEGIN
  FOR x IN 1..15
  LOOP
    IF Substr(telefone, x, 1) IN ('0', '1', '2', '3','4', '5','6', '7','8', '9')
    THEN s := To_Number(Substr(telefone, x, 1));
        soma := soma + s;
    END IF;
  END LOOP;
  RETURN soma;
END soma_numeros;
```

Exemplo

```
set serveroutput on;
DECLARE
    telefone VARCHAR(15):= '(016) 3373-3373';
    soma NUMBER(4);
BEGIN
    dbms_output.put_line ( 'O telefone de entrada é: '
                          || telefone );
    soma := soma_numeros (telefone);
    dbms_output.put_line ( 'A soma dos dígitos do
                          telefone é: ' || soma );

END;
```

Functions

- Execução

```
SELECT nome_func (lista_de_parâmetros) FROM DUAL;
```

- Remoção do banco de dados

```
DROP FUNCTION nome_func;
```

FROM DUAL

- Tabela adicional criada pelo Oracle
 - uma coluna: dummy
 - um registro: X
- Características
 - proprietário: SYS
 - visível para qualquer usuário
- Funcionalidade
 - útil para selecionar pseudo-colunas