FIAP
A MELHOR FACULDADE DE TECNOLOGIA

Subprogramas (Procedures)

Disciplina
Programação em Banco de Dados
Leonardo Uchida / Jorge Surian

Referências Bibliográficas
Manuais ou Materiais oficiais da Oracle ou Oracle Press

*Oracle9i / 10g PL/SQL * Guia de Consulta Rápida – Celso Henrique Poderoso*

Este material é apenas um guia de estudo e não substitui a leitura da referência bibliográfica e a consulta de anotações de sala de aula

Subprogramas

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *São blocos PL/SQL nomeados que podem receber parâmetros e ser invocado*
- *Conjunto de blocos que deve atender determinada regra de negócio e nomeado de forma significativa*
- *Pode ser "chamado" por parâmetros*
- *Interage com qualquer linguagem de programação que consiga conectar com o Oracle Database*
- *Existem 2 tipos: Stored Procedure e Functions*
- *Geralmente "Stored Procedure" é usada para executar uma ação e a "function" para computar/selecionar e retornar um valor*

Stored Procedure

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Esse conjunto de blocos PL/SQL necessita ser chamado para ser executado*
- *Está vinculado a um owner*
- *Vantagens:*
  - *Reusabilidade de código nas aplicações regra sobre regra*
  - *Portabilidade (plataformas)*
  - *Aumento de performance (cliente x servidor)*
  - *Manutenção centralizada*

Stored Procedure: Sintaxe

```
[CREATE [OR REPLACE]]
PROCEDURE procedure_name[(parameter[, parameter]...)]
   [AUTHID {DEFINER | CURRENT_USER}] {IS | AS}
   [PRAGMA AUTONOMOUS_TRANSACTION;]
   [local declarations]
BEGIN
   executable statements
[EXCEPTION
   exception handlers]
END [name];
```

- *Sintaxe para Criação*

  CREATE [OR REPLACE] PROCEDURE nome

  [(parâmetro, ...)] {IS|AS}

  [declarações]

  BEGIN

  <comandos>

  END [nome];

- *Sintaxe para Eliminar*

  DROP PROCEDURE nome;

Stored Procedure: Parâmetros

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Os parâmetros devem ser declarados com a cláusula IN, IN OUT ou OUT*

- *Parâmetros :*
  - ***IN** => Seu conteúdo não pode ser modificado. Esse é o parâmetro DEFAULT caso não seja declarado. Somente é possível fazer **Read** desse tipo de parâmetro. A cláusula DEFAULT pode ser associada a esse tipo de parâmetro*
  - ***IN OUT** => Tem um conteúdo e pode ser modificado, caso necessite. Pode ser utilizado como **Read** e **Write** dentro do subprograma. A cláusula DEFAULT não pode ser associada a esse tipo de parâmetro*
  - ***OUT** => Não tem conteúdo, e pode ser atribuído conteúdo dentro do procedimento. Somente pode ser utilizado para atribuir valor. Não é possível fazer acesso **Read**, somente **Write**. A cláusula DEFAULT não pode ser associada a esse tipo de parâmetro*

Stored Procedure: Parâmetros

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

IN	OUT	IN OUT
The default	Must be specified	Must be specified
Passes values to a subprogram	Returns values to the caller	Passes initial values to a subprogram and returns updated values to the caller
Formal parameter acts like a constant	Formal parameter acts like an uninitialized variable	Formal parameter acts like an initialized variable
Formal parameter cannot be assigned a value	Formal parameter must be assigned a value	Formal parameter should be assigned a value
Actual parameter can be a constant, initialized variable, literal, or expression	Actual parameter must be a variable	Actual parameter must be a variable
Actual parameter is passed by reference (a pointer to the value is passed in)	Actual parameter is passed by value (a copy of the value is passed out) unless `NOCOPY` is specified	Actual parameter is passed by value (a copy of the value is passed in and out) unless `NOCOPY` is specified

Stored Procedure: Parâmetro Default

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

```
DECLARE
  emp_num NUMBER(6) := 120;
  bonus   NUMBER(6);
  merit   NUMBER(4);
  PROCEDURE raise_salary (emp_id IN NUMBER, amount IN NUMBER DEFAULT 100,
                          extra IN NUMBER DEFAULT 50) IS
    BEGIN
      UPDATE employees SET salary = salary + amount + extra
        WHERE employee_id = emp_id;
  END raise_salary;
BEGIN
  raise_salary(120); -- same as raise_salary(120, 100, 50)
  raise_salary(emp_num, extra => 25); -- same as raise_salary(120, 100, 25)
END;
/
```

Stored Procedure: exemplo 1

- *Criar uma procedure que insera Grupo de Locação dentro do Projeto Carloca*

```
create or replace procedure prc_insere_grupo(
p_cd_grupo in loc_grupo.cd_grupo%type,
p_ds_grupo in loc_grupo.ds_grupo%type,
p_vl_locacao_diaria in loc_grupo.vl_locacao_diaria%type
)  is
begin
begin
 INSERT INTO loc_grupo(cd_grupo, ds_grupo, vl_locacao_diaria)
 VALUES (p_cd_grupo, p_ds_grupo, p_vl_locacao_diaria) ;
 commit;
Exception
 when dup_val_on_index then
  raise_application_error( -20022, 'Erro chave primária! Insere Grupo ' || SQLErrM );
 when others then
  raise_application_error( -20023, 'Erro crítico ! Insere Grupo ' || SQLErrM );
end;
end;
```

Stored Procedure: execução

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Verficando os parâmetros da procedure*

  SQL> DESC prc_insere_grupo

- *Executando a procedure (exemplo 1)*

  SQL> execute PRC_INSERE_GRUPO(6,'Veículos Super-Esportivos', 678);

  (procedimento chamado por parâmetro posicional)

  OU

  SQL>execute PRC_INSERE_GRUPO(p_ds_grupo=>'Veículos Super-Esportivos', p_vl_locacao_diaria=>678, p_cd_grupo=>6);

  (procedimento chamado por parâmetro nomeado)

- *Verificando a informação*

  *SQL> SELECT * FROM LOC_GRUPO;*

Stored Procedure: Forma de Execução

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Parâmetro Posicional*
  - *Utilização dos parâmetros na ordem seqüencial de criação da procedure*
  - *Para os parâmetros que contém a cláusula DEFAULT, é opção facultativa sua utilização.*

- *Parâmetro Nomeado*
  - *Ordem de utilização arbitrária*
  - *Para os parâmetros que contém a cláusula DEFAULT, é opção facultativa sua utilização.*

Stored Procedure: exemplo 2

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

Insert com retorno do valor do código

CREATE SEQUENCE SEQ_LOCGRUPO start with 7 increment by 1 minvalue 1 nocache nocycle;

Stored Procedure: exemplo 2

```
create or replace procedure prc_insere_grupo(
p_ds_grupo          in   loc_grupo.ds_grupo%type,
p_vl_locacao_diaria    in   loc_grupo.vl_locacao_diaria%type,
p_cd_grupo          out loc_grupo.cd_grupo%type
)  is
begin
begin
   INSERT INTO loc_grupo(cd_grupo, ds_grupo, vl_locacao_diaria)
   VALUES (SEQ_LOCGRUPO.NEXTVAL, p_ds_grupo, p_vl_locacao_diaria)
   RETURNING cd_grupo INTO p_cd_grupo;
   commit;
exception
   when others then
            raise_application_error( -20003, 'Erro crítico ! Insere Grupo ' || SQLErrM
   );
end;
end prc_insere_grupo;
```

Stored Procedure: exemplo 2

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- ***Executando a procedure (1ª opção)***

```
set serveroutput on
declare v_codigo number;
begin
        prc_insere_grupo('Motos maiores que 1000 cilindradas', 864, v_codigo);
    dbms_output.put_line('Valor do Código=>' || v_codigo);
end;
```

- ***Executando a procedure (2ª opção): Referência Nome Parâmetro***

```
set serveroutput on
declare v_codigo number;
begin
prc_insere_grupo(p_cd_grupo=>v_codigo, p_vl_locacao_diaria=>864,
p_ds_grupo=>'Motos maiores que 1000 cilindradas');
 dbms_output.put_line('Valor do Código=>' || v_codigo);
end;
```

Stored Procedure

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

SQL> DESC PRC_INSERE_GRUPO -- Exibe os parâmetros

SQL> SHOW ERRORS; -- Exibe os erros existentes

Stored Procedure

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Para RECOMPILAR uma procedure :*

  SQL> ALTER PROCEDURE PRC_INSERE_GRUPO COMPILE;

Stored Procedure: Privilégio de execução

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Por default, procedimentos armazenados e métodos SQL embutidos são executados pelo privilégio do proprietário do objeto.*
- *O privilégio utilizado para garantir a execução da procedure é o comando EXECUTE.*
  - *Exemplo: GRANT EXECUTE ON <objeto> TO <usuário/role>;*
- *O privilégio utilizado para retirar o privilégio de execução da procedure é o comando REVOKE.*
  - *Exemplo: REVOKE EXECUTE ON <objeto> FROM <usuário/role>;*
- *O usuário que irá executar não precisa ter privilégios sobre os objetos que fazem parte do procedimento.*
- *Porém, com a clásula AUTHID é permitido que a execução do procedimento siga os privilégios definidos do usuário executor e não do proprietário do objeto*

Stored Procedure: AUTHID current user

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

```
CREATE OR REPLACE PROCEDURE create_dept (
   v_deptno NUMBER,
   v_dname  VARCHAR2,
   v_mgr    NUMBER,
   v_loc    NUMBER)
AUTHID CURRENT_USER AS
BEGIN
   INSERT INTO departments VALUES (v_deptno, v_dname, v_mgr, v_loc);
END;
/
CALL create_dept(44, 'Information Technology', 200, 1700);
```

Outras considerações

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

- *Depois da clausula IS da procedure, as variáveis criadas são locais;*

- *Declaração de argumentos na procedure não se utiliza o tamanho máximo, apenas seu tipo de dado;*

- *Após compilar a procedure e aparecer erro no momento da compilação o codigo fonte e o erro são armazenados no dicionário de dados. As visões USER_SOURCE e USER_ERRORS respectivamente fornecem o acesso a essas informações;*

Subprograma local dentro de uma Procedure

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

```
create or replace procedure prc_gera_volume_vendas as

v_novo_nome varchar2(100);

function fun_gera_novo_nome( v_nome in varchar2) return varchar2 is
v_nome_saida varchar2(100);
begin
   select REGEXP_REPLACE( v_nome, '(.)', '\1 ') into v_nome_saida from dual;
   return v_nome_saida;
end fun_gera_novo_nome;

begin
   for x in (     select nm_func, vl_salario, vl_perc_comissao
                  from loc_funcionario )
   loop
         v_novo_nome := fun_gera_novo_nome( x.nm_func);
         dbms_output.put_line('Novo Nome Func: ' || v_novo_nome);
   end loop;

   for y in (     select nm_cliente, nr_estrelas, telefone
                  from loc_cliente  )
   loop
         v_novo_nome := fun_gera_novo_nome( y.nm_cliente);
         dbms_output.put_line('Novo Nome Cliente: ' || v_novo_nome);
   end loop;

end prc_gera_volume_vendas;
```

FIAP
A MELHOR FACULDADE DE TECNOLOGIA

Não basta saber, é preciso exercitar.

Lista de Exercícios