



## PROCEDURES E FUNÇÕES EM PL/SQL

Publicado por: Peterson Ferreira dos Santos 28 de dezembro de 2008 em Artigos 1 Comentário

Bom galera a pedido, está o tutorial sobre **PL/SQL** de uma maneira bem simples, utilizando a linguagem PL/SQL para fazer *Procedures*, e *Functions*.

### Procedures

Uma procedure nada mais é do um bloco *PL/SQL* nomeado. A grande vantagem sobre um bloco *PL/SQL* anônimo é que pode ser compilado e armazenado no banco de dados como um objeto de *schema*. Graças a essa característica as procedures são de fácil manutenção, o código é reutilizável e permitem que trabalhemos com módulos de programa.

Uma procedure é, então, um bloco *PL/SQL* nomeado que pode aceitar argumentos (também chamado de parâmetros) e pode ser chamada por um programa, uma sessão *SQL* ou uma *trigger*.

Durante a instalação do banco de dados **Oracle** um script é executado automaticamente e cria toda a estrutura necessária para que as procedures sejam executadas. Eventualmente esse procedimento automático pode falhar devido a alguma falha física no disco rígido, nesse caso o usuário *SYS* pode recriar a estrutura através do script *SQL DBMSSTDY.SQL*.

Para criar uma procedure o usuário precisa ter o privilégio de sistema *CREATE PROCEDURE*, para criar a procedure em outros *schemas* o usuário deve ter o privilégio de *CREATE ANY PROCEDURE*. Este é um ponto muito interessante sobre as procedures, os privilégios para criação de procedures têm que ser concedidos explicitamente, ou seja, não pode ser adquirido através de roles.

Para executar uma procedure externa é necessário ter o privilégio de *EXECUTE*. Caso queira alterar a procedure de outro *schema* deve ter o privilégio de sistema *ALTER ANY PROCEDURE*.

### VÍDEOS

Os Cursos Tecnólogos são...



A sintaxe básica de uma procedure é:

```
1. CREATE [OR REPLACE] PROCEDURE [schema.]nome_da_procedure
2. [(parâmetro1 [modo1] tipodedado1,
3.     parâmetro2 [modo2] tipodedado2,
4.     ...)]
5. IS|AS
6. Bloco PL/SQL
```

Onde:

**REPLACE** – indica que caso a procedure exista ela será eliminada e substituída pela nova versão criada pelo comando;

**BLOCO PL/SQL** – inicia com uma cláusula *BEGIN* e termina com *END* ou *END nome\_da\_procedure*;

**NOME\_DA\_PROCEDURE** – indica o nome da *procedure*;

**PARÂMETRO** – indica o nome da variável *PL/SQL* que é passada na chamada da *procedure* ou o nome da variável que retornará os valores da procedure ou ambos. O que irá conter em parâmetro depende de *MODO*;

**MODO** – Indica que o parâmetro é de entrada (*IN*), saída (*OUT*) ou ambos (*IN OUT*). É importante notar que *IN* é o modo *default*, ou seja, se não dissermos nada o modo do nosso parâmetro será, automaticamente, *IN*;

**TIPODEDADO** – indica o tipo de dado do parâmetro. Pode ser qualquer tipo de dado do *SQL* ou do *PL/SQL*. Pode usar referencias como *%TYPE*, *%ROWTYPE* ou qualquer tipo de dado escalar ou composto. Atenção: não é possível fazer qualquer restrição ao tamanho do tipo de dado neste ponto.

**IS|AS** – a sintaxe do comando aceita tanto *IS* como *AS*. Por convenção usamos *IS* na criação de procedures e *AS* quando estivermos criando pacotes.

**BLOCO PL/SQL** – indica as ações que serão executadas por aquela *procedure*.

Vamos ver um exemplo de procedure para ajudar nosso entendimento:

```
1. CREATE OR REPLACE PROCEDURE aumenta_sal (p_empno IN emp.empno%TYPE) IS
2. BEGIN
3.     UPDATE
4.         scott.emp
```

```

5.         SET
6.             sal = sal * 1.10
7.         WHERE
8.             empno = p_empno;
9. END aumenta_sal;
10. /

```

Neste exemplo estamos criando uma procedure para aumentar o salário de um funcionário em 10%. A primeira linha define o NOME DA PROCEDURE, que vai ser **AUMENTA\_SAL**.

A linha dois define o parâmetro **P\_EMPNO** no modo *IN*. Ou seja, vai ser um dado informado na chamada da procedure. Em seguida determinamos que ele será do mesmo tipo e tamanho que a coluna **EMPNO** da tabela **EMP**. Isso é feito através da referencia **EMP.EMPNO%TYPE**.

Podemos verificar o estado de nossa procedure através de uma simples consulta:

```

1. SELECT object_name, status
2.     FROM user_objects
3.     WHERE object_name LIKE '%AUMENTA%';

```

```

1. SELECT empno, sal
2.     FROM scott.emp;

```

```

1. EMPNO      SAL
2. -----
3. 7839      5000
4. 7698      2850
5. 7782      2450

```

```

1. CALL AUMENTA_SAL(7839);

```

Ou

```

1. EXECUTE AUMENTA_SAL(7839);

```

```

1. SELECT empno, sal
2.     FROM scott.emp;

```

```

1.      EMPNO      SAL
2. -----
3.       7839      5500
4.       7698      2850

```

```
5.      7782      2450
```

Podemos notar que o salário do funcionário 7839 aumentou em 10%.

## Funções

### Sintaxe Básica:

```
1. CREATE [OR REPLACE] FUNCTION nome_da_função
2. [( parameter1 [ mode1] datatype1,
3. parameter2 [ mode2] datatype2,
4. . . .)]
5. RETURN tipo_de_dado
6. IS|AS
7. Bloco PL/SQL;
```

Ao contrário das *procedures* as funções tem que retornar ao menos um valor.

```
1. CREATE OR REPLACE FUNCTION pega_sal
2.     (p_id IN emp.empno%TYPE)
3.     RETURN NUMBER
4.     IS
5.     v_sal emp.sal%TYPE :=0;
```

Artigos ▾ | Códigos | Notícias ▾ | Dicas | Ferramentas ▾ | Vídeos | #Games

Pesquisa...

Ir

```
9.     FROM scott.emp
10.    WHERE empno = p_id;
11.    RETURN v_sal;
12. END pega_sal;
13. /
```

## Executando

```
1. VARIABLE g_sal NUMBER
2.
3. EXECUTE :g_sal := pega_sal(7839)
4.
5. PRINT g_sal
```

## Função para calcular CPMF

```
1. CREATE OR REPLACE FUNCTION cpmf(p_value IN NUMBER)
2.     RETURN NUMBER IS
3. BEGIN
4.     RETURN (p_value * 0.038);
5. END cpmf;
6. /
7. SELECT empno, ename, sal, cpmf(sal)
8. FROM scott.emp
9. WHERE deptno = 10;
```

0

COMPARTILHE! Tweet Like 6 G+ 1 Share <http://codft.me/2pX>

Marcados com: BANCO DE DADOS ORACLE PL/SQL PROCEDURES SQL

## QUEREMOS SABER SUA OPINIÃO

2 Comments

Sort by **Oldest**



Add a comment...



**Bruno Solé** · USTP (Universidade de STP)

legal!

Like · Reply · Jul 13, 2015 2:21pm



**Rica Neto** · Reporter at Jornal o Parvo

fine

Like · Reply · Jan 25, 2016 8:03am

Facebook Comments Plugin

## @SIGA-ME

Super Mario Run já rende frutos à Nintendo: ações da empresa disparam na Bolsa! <https://t.co/hfvbZ9vfcz> 17 minutos ago

#Games de tudo que a Apple revelou <https://t.co/7d4uDEPD4N>

Apple reduz os preços dos iPhones no Brasil, após anúncio de ontem: <https://t.co/HJ8VijFJIU> 1 hora ago

Um dos maiores portais da Rússia foi invadido e criminosos levaram quase 100 MILHÕES de credenciais! <https://t.co/tXKGxsMT1n> 2 horas ago

Na briga dos consoles, Microsoft vai aceitar seu Xbox 360 usado na compra de um Xbox One. Entenda: <https://t.co/Ziyop4I3XM> 2 horas ago

para pesquisar digite e aperte enter

## TAGS

ABAP Android Aplicativo Aplicativos App  
Apple atualização Brasil Diversos Efe  
Facebook Firefox Games  
Google  Pesquisa... Ir

iPhone JavaScript Jogo Jogos Justiça  
Mercado Microsoft Música Negócios PHP  
Privacidade Rede Social Redes Sociais  
Rumor Samsung SAP Segurança  
Smartphone Sony Tecnologia Trailer  
Twitter videos Windows Windows 10 Xbox  
One Youtube

## SE LIGUE NO CÓDIGO FONTE



Apelido

.....

Artigos Estilos Códigos Notícias Dicas Ferramentas Vídeos

[Cadastre-se](#)

[Perdeu a senha?](#)

- ### SOBRE
- [Ajuda](#)
  - [Seja um colaborador](#)
  - [Fale conosco](#)
  - [Sobre](#)
  - [Anuncie](#)