ITB – INSTITUTO TECNOLÓGICO DE BARUERI

LECO – LINGUAGEM ESTRUTURADA DE CONSULTA

PROFESSORA VIVIANE

STORED PROCEDURE OU PROCEDIMENTOS ARMAZENADOS

Stored Procedures (SPs) ou Procedimentos Armazenados são blocos PL/SQL nomeados, também chamados de subprogramas, que são armazenados como objetos do banco de dados.

As SPs permitem criar blocos de código que podem conter instruções procedurais e instruções SQL e podem possuir, opcionalmente, parâmetros de entrada e/ou saída. Esses blocos possuem um nome que permite identificá-los no banco de dados. As aplicações e outros blocos PL/SQL podem invocar ou utilizar as SPs através de seu nome. Elas são ótimas para otimizar performance e aumentar a segurança das aplicações, além de promoverem

CRIAÇÃO OU ALTERAÇÃO DE STORED PROCEDURE

SINTAXE

```
CREATE [OR REPLACE] PROCEDURE proc_name [list of parameters]
 IS
    Declaration section
 BEGIN
    Execution section
 EXCEPTION
   Exception section
 END;
```

EXEMPLO DE STORED PROCEDURE

Atualiza o preço de um livro em 10%:

Criando ou alterando a procedure:

```
CREATE OR REPLACE PROCEDURE aumenta_preco
(titulo_livro IN livro.titulo%TYPE) IS
BEGIN
UPDATE LIVRO SET preco = preco * 1.10
WHERE titulo = titulo_livro;
END aumenta_preco;
```

Executando a procedure: CALL AUMENTA_PRECO('ABC');

EXEMPLO DE STORED PROCEDURE

Neste exemplo estamos criando uma procedure para aumentar o preço de um livro em 10%. A primeira linha define o NOME DA PROCEDURE, que vai ser AUMENTA_PRECO.
A linha dois define o parâmetro titulo_livro 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 TITULO da tabela LIVRO. Isso é feito através da referencia LIVRO.TITULO%TYPE.

CRIAÇÃO OU ALTERAÇÃO DE STORED PROCEDURE

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;

CRIAÇÃO OU ALTERAÇÃO DE STORED PROCEDURE

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.

CRIAÇÃO OU ALTERAÇÃO DE STORED PROCEDURE

- Parâmetros IN – passamos o valor na própria procedure.
- Parâmetros OUT – recebemos o valor a partir da chamada de blocos externos.
- Parâmetros IN OUT – passamos um valor inicial para a procedure e recebemos de volta uma atualização.

EXERCÍCIOS

1. Crie uma stored procedure que inclui uma nova categoria

FUNCTIONS OU FUNÇÕES

Functions ou Funções (em português) também são blocos PL/SQL nomeados, muito semelhantes as Stored Procedures. A diferença principal é que as Funções obrigatoriamente devem retornar um valor.

As funções são muito utilizadas para computar valores, promover reusabilidade e facilidade de manutenção e podem ser chamadas a partir de outros blocos PL/SQL ou até mesmo por instruções SQL. É muito comum utilizarmos funções em instruções SQL para efetuar conversões de dados, formatar datas, contar o total de linhas, etc.

CRIAÇÃO OU ALTERAÇÃO DE FUNÇÃO

SINTAXE

```
CREATE [OR REPLACE] FUNCTION function_name
  [(parameter_name [IN | OUT | IN OUT] type [, ...])]
  RETURN return_datatype
  {IS | AS}
  BEGIN
     < function_body >
  END [function_name];
```

CRIAÇÃO OU ALTERAÇÃO DE FUNÇÃO

- CREATE [OR REPLACE] FUNCTION: Caso uma function já exista com o mesmo nome, ela será reescrita devido ao termo ‘replace’. Caso contrário, ela será criada de acordo com o termo ‘create’.

- Function_name: Será o nome atribuído para essa função.

- Parameters: a lista opcional de parâmetros contém os nomes, os modos e os tipos que esses parâmetros terão. O IN representa o valor que será passado de fora, já o OUT representa que este parâmetro será utilizado para retornar um valor de fora do procedimento.

CRIAÇÃO OU ALTERAÇÃO DE FUNÇÃO

- Return_datatype: é o tipo de retorno que será utilizado, sendo este SQL ou PL/SQL. Podemos neste caso utilizar referências como o %TYPE ou %ROWTYPE se necessário, ou mesmo utilizar qualquer tipo de dados escalar ou composto.

- IS/AS: por convecção, temos o ‘is’ para a criação de funções armazenadas e o ‘as’ quando criamos pacotes (packages).

- function_body: contém o bloco PL/SQL que inicia com a claúsula BEGIN e finaliza com END [function_name], e executa neste momento todas as instruções necessárias.