PL/SQL – Parte 4

Curso de Introdução a Oracle 11g: SQL/Avançado

Prof.: Marlon Mendes Minussi
marlonminussi@gmail.com

Packages

- É uma coleção de objetos de banco de dados, r componentes de um sistema em módulos.como stored procedures, funções, variáreis, constantes e cursores.
- Ele contem subprogramas que podem ser chamados a partir de um trigger, uma procedure ou uma function.
- Um package é um grande aliado do desenvolvedor, pois permite organizar melhor componentes de um sistema em módulos.

Packages

- Por exemplo, um package chamado aumenta_conta_pac_pkg pode conter funções e stored procedures relacionados com a tabela pacientes.

- O package melhora o desempenho da máquina, já que ele transfere para a memória diversos objetos de uma vez.

Packages

- Packages são compostos de duas partes(seções) que precisam ser criadas separadamente por dois comandos distintos:
- A especificação e o corpo do pacote.
- A seção de especificação funciona como espécie de sumário do conteúdo do corpo do package. Fisicamente são objetos distintos. Nela são declarados os tipos, variáveis, constantes, exceções, cursores e subprogramas.
- No corpo do pacote é concluída a definição dos procedimentos, funções e outros objetos e feita a implementação da especificação.

Packages

- Uma PACKAGE é constituída de duas partes.

- A primeira é a PACKAGE SPECIFICATION e a segunda é a PACKAGE BODY, onde através delas podemos definir a função do nosso "pacote" dentro do banco de dados.

Package Specification

- O comando CREATE PACKAGE
- Cria a especificação para um package armazenado. Como vimos anteriormente, um package é uma coleção encapsulada de procedimentos, funções e outros objetos de programas relacionados ou armazenados em conjunto no banco de dados.

- Sintaxe:

  CREATE [OR REPLACE] PACKAGE [schema.] nome

  {IS|AS} pl/sql_package_spec

Package Body

- O comando CREATE PACKAGE BODY
- Cria o corpo de um package armazenado. Package é uma coleção encapsulada de procedimentos, funções e outros objetos de programa, relacionados ou armazenados em conjuntos no banco de dados.

- Sintaxe:

  CREATE [OR REPLACE] PACKAGE BODY [schema.] nome

  {IS|AS} pl/sql_package_body

Sintaxe

```
CREATE [OR REPLACE]
PACKAGE <nome do pacote> AS/IS                    — parte da especificação
                <declarações>
END [<nome do pacote>];

CREATE [OR REPLACE]
PACKAGE BODY <nome do pacote> AS/IS      — corpo do pacote
                        <corpo dos subprogramas>
[BEGIN
                        <comandos de inicialização>]
END [<nome do pacote>];
```

Criando um Package Specification

- Ex.:

```
CREATE OR REPLACE
PACKAGE atualiza_desconto_pkg IS
    PROCEDURE aumenta_desconto1 (p_cod_atendimento IN
    atendimento.cod_atendimento%TYPE);
    PROCEDURE aumenta_desconto_todos;
END atualiza_desconto_pkg;
/
```

- No próximo slide veja o conteúdo do corpo desse pacote. Ele possui duas procedures. A primeira toma um argumento, cod_atendimento para que seja feito um aumento de desconto para o referido atendimento. A segunda procedure fará uma aumento de todos os descontos.

Criando um Package Body

- Ex.:

```
CREATE OR REPLACE
PACKAGE BODY atualiza_desconto_pkg IS
        PROCEDURE aumenta_desconto1 (p_cod_atendimento IN
    atendimento.cod_atendimento%TYPE) IS
        BEGIN
          UPDATE atendimento
          SET desconto=NVL(desconto,10)*1.1
          WHERE cod_atendimento=p_cod_atendimento;
        END aumenta_desconto1;
        PROCEDURE aumenta_desconto_todos IS
        BEGIN
          UPDATE atendimento
          SET desconto=desconto*1.1;
        END aumenta_desconto_todos;
END atualiza_desconto_pkg;
/
```

Package

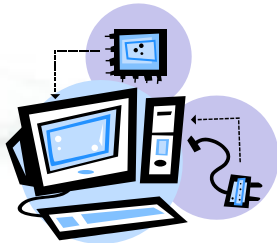
- Para acessar uma função através de um pacote você pode utilizar os seguintes comandos:

```
EXECUTE  atualiza_desconto_pkg.aumenta_desconto1(8)

BEGIN
   atualiza_desconto_pkg.aumenta_desconto_todos;
END;
```

Atividade 1

- Baseado no exemplo anterior crie um novo pacote só que agora ao invés de duas procedures use uma procedure e uma function.
- A procedure deve receber dois argumentos, o primeiro é o cod_pac e o segundo percentual de aumento, e então deve fazer o calculo do aumento do salário do paciente conforme o valor digitado na execução do pacote.
- A Função deve retornar a quantidade de pacientes cadastrados em um determinado estado, deve ser declarado um paramentro estado que retornará um number para um variável total_pac.
- Para acessar uma função em um pacote você pode utilizar os seguintes comandos:
  - SELECT aumenta_conta_pac_pkg.conta_pac('RS') FROM dual;



Excluindo um Package

- Quando um pacote é excluído, todas as referencias feitas a ele e seus objetos se tornarão inválidos.
- Portanto, tome cuidado antes de fazer isso. Se ele é especificado dentro de um trigger, o Oracle tenta recompilar o package, caso não encontre. É emitido um erro.
- Sintaxe:
  - DROP PACKAGE [BODY] nome;
- A opção body exclui o corpo do package e deixa a seção de especificação intacta. Se for omitida esta opção, as duas seções do package serão excluídas.

Excluindo um Package

- O package inteiro é configurado para a memória quando a procedure dentro dele é chamada pela primeira vez.
- O corpo do package pode ser substituído e recompilado sem afetar sua própria especificação. Isto permite que os objetos de esquema que fazem referência aos elementos do package não precisem ser recompilados, a não ser que a especificação também seja substituída.
- Por exemplo poderia trocar a tabela atendimento por exame no procedimento aumenta_desconto_todos, pois ambas tabelas possuem o atributo desconto. Poderíamos apagar o corpo do package alterar e recompilar.
- Ex.:

```
DROP PACKAGE  BODY atualiza_desconto_pkg;
```

Recompilando um Package

- Recompilar um package uma recomendação interessante e relevante, pois, fazendo isso, sem dúvida, melhora o desempenho e evita problemas eventuais durante a execução de alguma tarefa.
- Para recompilar uma das partes do package, use o comando ALTER PACKAGE.
- Você pode recompilar somente o corpo usando a opção de body do comando:

```
ALTER PACKAGE atualiza_desconto_pkg COMPILE BODY;
ALTER PACKAGE atualiza_desconto_pkg COMPILE;
```

Packages prontos do Oracle

- O Oracle possui dezenas de packages.
- Um deles é o package DBMS_OUT possui diversas procedures que permitem o envio de mensagens a partir de outras procedures, packages ou triggers.
- Durante a fase de criação e teste de um trigger, podemos usar o package DBMS_OUTPUT para verificar o conteúdo de colunas.

Packages prontos do Oracle

- Normalmente, os packages possuem diversas procedures prontas responsáveis pela suas tarefas. Veja as contidas em DBMS_OUTPUT:

Rotina	Descrição
ENABLE	Ativa a exibição de mensagens
DISABLE	Desativa a exibição de mensagens
PUT e PUT_LINE	Coloca uma linha no buffer
NEW_LINE	Termina a linha criada com PUT
GET_LINE e GET_LINES	Recupera a informação que está no buffer

Packages prontos do Oracle

- Veja um exemplo do uso desse package, chamado por uma trigger disparado quando qualquer operação DML é executada na tabela paciente.

```
CREATE OR REPLACE TRIGGER saldif
BEFORE DELETE OR INSERT OR UPDATE ON paciente
FOR EACH ROW
WHEN (new.cod_pac>0)
DECLARE
    sal_dif number;
BEGIN
    sal_dif:= :new.salario - :old.salario;
    DBMS_OUTPUT.PUT( 'ANTIGO: '|| :old.salario);
    DBMS_OUTPUT.PUT( ' NOVO: '|| :new.salario);
    DBMS_OUTPUT.PUT_LINE( ' DIFERENÇA ' || sal_dif);
END;
/
UPDATE paciente SET salario=salario*1.3 WHERE cod_pac=8;
```

Atividade 2

```
CREATE OR REPLACE
PACKAGE med_package AS
    PROCEDURE admissao (cod_med IN medico.cod_med%TYPE, nome IN medico.nome%TYPE, crm IN medico.crm%TYPE, especialidade IN medico.especialidade%TYPE, fone IN medico.fone%TYPE, salario IN medico.salario %TYPE);
END med_package;
```

Atividade 2

```
CREATE OR REPLACE
PACKAGE BODY med_package As
PROCEDURE admissao (cod_med IN medico.cod_med%TYPE, nome IN
    medico.nome%TYPE, crm IN medico.crm%TYPE, especialidade IN
    medico.especialidade%TYPE, fone IN medico.fone%TYPE, salario IN
    medico.salario%TYPE) IS   BEGIN
    INSERT INTO medico (cod_med, nome, crm, especialidade, fone, salario)
    VALUES (cod_med, nome, crm, especialidade, fone, salario);
   END admissao;
END med_package;
```

Atividade 2

- Para testar :

```
BEGIN
   med_package.admissao(5,'TESTE', 16164547, 'TESTE', 895865478,
    8500.45);
END;
```

- Agora você deve alterar a Specification e Body do Package adicionando uma nova procedure chamado demissão que exclua um registro da tabela médico através da passagem do parâmetro cod_med.

Referências a Packages

- Para fazer uma referência aos tipos, objetos e subprogramas declaradas dentro da especificação de packages deve-se usar a seguinte notação:
  - <nome de package>. <tipo>
  - <nome de package>. <objeto>
  - <nome de package>. <subprograma>
- Pode-se fazer referência a pacotes de dentro de database triggers, stored subprograms e blocos PL/SQL dentro de programas e blocos PL/SQL anônimos.
- A inicialização da parte executável do package é feita uma única vez, na primeira vez que for feita referência ao package (**por sessão**).

Bibliografia

- RAMALHO, José Antonio. *Oracle 10g. São Paulo : Pioneira Thomson Learning, 2005.*