

Packages

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

- *Após o término desta lição, você estará habilitado para :*
 - *Descrever o uso de Packages e listar seus componentes*
 - *Criar Packages para agrupar variáveis, constantes, exceptions, procedures e functions*
 - *Invocar Packages*
 - *Remover Packages*

- *É uma coleção de elementos PL/SQL que são empacotados com uma sintaxe de Begin – End, parecido com um meta-bloco.*
- *Packages são objetos do banco de dados, equivalentes a bibliotecas que agrupam e armazenam:*
 - *Procedures*
 - *Functions*
 - *Definições de Cursores*
 - *Variáveis e constantes*
 - *Definições de Exceções*
 - *PL/SQL Table e declarações TYPE*

- *É composta de 2 partes*
 - *Especificação (Specification)*
 - *Corpo (Body)*



Composição de um Package

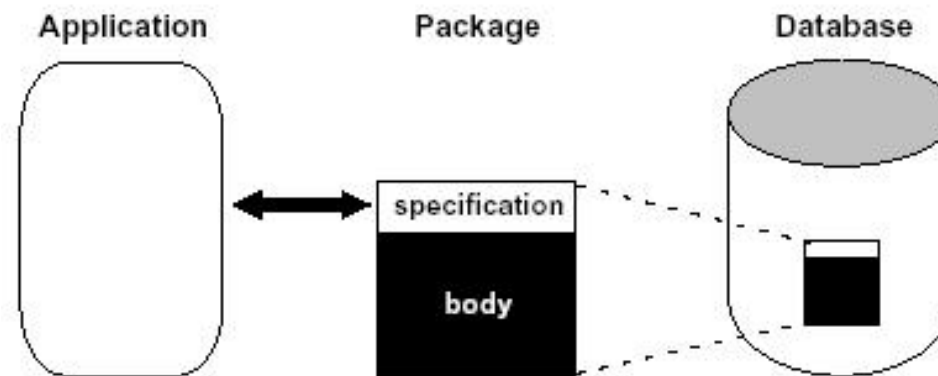
- *Specification*

- *Nesta área são feitas as declarações públicas, ou seja, as variáveis, constantes, exceções e subprogramas que estarão disponíveis para uso externo à package.*
- *É a interface para suas aplicações e todos os componentes disponíveis para uso e sua respectiva definição de uso estão definidas aqui.*

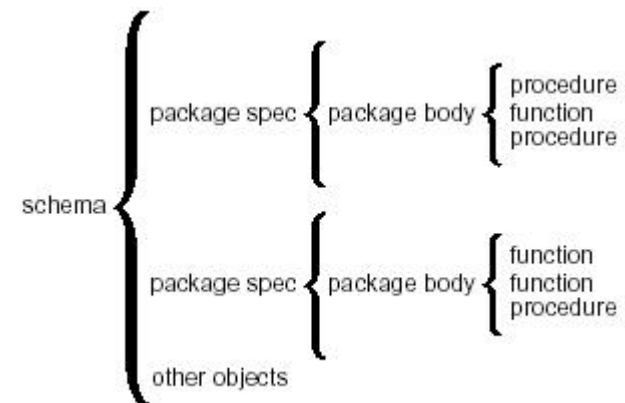
- *Sintaxe de criação :*

```
CREATE OR REPLACE PACKAGE <nome_package> IS  
    PROCEDURE <Nome_Procedure>;  
    FUNCTION <Nome_Function>;  
    Declaração de variáveis, constantes, exceções e cursores  
END;
```

Composição de um Package



*Entendendo o objeto
Package*



CREATE OR REPLACE PACKAGE PKG_CARLOCA IS

*PROCEDURE PRC_INCLUI_GRUPO(P_DS_GRUPO VARCHAR2,
P_VALOR NUMBER);*

END PKG_CARLOCA;

- *Acima criamos um Pacote contendo 1 objeto : Uma procedure*
- *Este pacote está disponível e poderá ser utilizado apenas pelos usuários que irão receber o privilégio de acesso, ou o próprio owner.*

- *Body(Corpo)*
 - *Nesta área são feitas as declarações privadas, que estarão disponíveis apenas dentro da própria package, e a definição das ações para os subprogramas.*

- *Sintaxe de criação :*

```
CREATE OR REPLACE PACKAGE BODY <nome_package> IS  
    PROCEDURE Nome_Procedure( lista de parametros) IS  
    BEGIN  
        ...  
    END;  
END;
```

Exemplo de um Package Body

```
CREATE OR REPLACE PACKAGE BODY PKG_CARLOCA IS

    PROCEDURE PRC_INCLUI_GRUPO( P_DS_GRUPO IN VARCHAR2,
                                P_VALOR NUMBER ) IS
        V_CD_GRUPO LOC_GRUPO.CD_GRUPO%TYPE;
    BEGIN
        SELECT NVL( MAX( CD_GRUPO ), 0 ) + 1 INTO V_CD_GRUPO
        FROM LOC_GRUPO;
        BEGIN
            INSERT INTO LOC_GRUPO (CD_GRUPO, DS_GRUPO, VL_LOCACAO_DIARIA)
            VALUES ( V_CD_GRUPO, P_DS_GRUPO, P_VALOR );
        EXCEPTION
            WHEN OTHERS THEN
                RAISE_APPLICATION_ERROR( -20003, 'Erro Inserir Grupo! ' || SqlErrM );
        END;
    END PRC_INCLUI_GRUPO;

END PKG_CARLOCA;
```

- *Execução de uma package :*
execute package.objeto(<parâmetros>...)
- *Recompilar um package*
ALTER PACKAGE <nome_pacote> COMPILE;
ALTER PACKAGE <nome_pacote> COMPILE BODY;
- *Eliminar uma package*
DROP PACKAGE <nome_pacote>

Exemplo de execução de um Package

- Executar a procedure *prc_inclui_regiao* que está dentro do pacote *PKG_CARLOCA*

```
EXECUTE PKG_CARLOCA.PRC_INCLUI_GRUPO('ÔNIBUS EXECUTIVO',1000);
```

Package: declarando variáveis públicas

create or replace package global_var is

v_milhas_2_kilom CONSTANT number := 1.6093;

v_jardas_2_metros CONSTANT number := 0.9144;

end global_var;

Privilégio de acesso

Grant execute on global_var to public;

Executando

EXECUTE DBMS_OUTPUT.PUT_LINE('20 Milhas = ' || 20
global_var.v_milhas_2_kilom || ' km');*

- **Organização**

Agrupa procedures / functions

- **Segurança**

Acesso para todos os objetos da package uma única vez

Apenas as declarações na package specification são visíveis para as aplicações

- **Desempenho**

Carrega toda a estrutura da package em memória na primeira chamada

Reduz acesso a disco em chamadas subsequentes

Apenas uma cópia na memória para todos os usuários

- **DBMS_ALERT**
Avisa determinada aplicação por meio de database triggers quando específicos eventos no banco de dados ocorrem
- **DBMS_OUTPUT**
Habilita enviar mensagens de texto e conteúdos de variáveis para ser possível realizar um debug inicial
- **DBMS_PIPE**
Permite diferentes sessões se comunicarem por meio de nomes específicos de pipes. Um pipe é uma área de memória usada por um processo para passar informação para outro processo
- **UTL_FILE**
Permite ler e gravar operações de arquivos textos realizados no SO.
- **UTL_HTTP**
Permite que seus programas PL/SQL executem procedimentos feitos em HTTP
- **UTL_SMTP**
Permite que seus programas PL/SQL enviem emails eletrônicos pelo protocolo SMTP, com possibilidade de anexar arquivos

- *Os pacotes podem ser administrados da mesma forma que as procedures/functions*
- *Documentação => dentro da visão USER_OBJECTS e USER_SOURCE*
- *Erros de Compilação => Show Error*

Você aprendeu...

- *Descrever o uso de Packages e listar seus componentes*
- *Criar Packages para agrupar variáveis, constantes, exceptions, procedures e functions*
- *Invocar Packages*
- *Remover Packages*

Não basta saber, é preciso exercitar.

Lista de Exercícios

Copyright © 2013

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, dos Professor (autor).