
FIAP

Sistemas de Informação

Prof. Me. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

Criação de Pacotes

Objetivos

- Definir o que é um pacote e listar os seus componentes
- Criar um pacote
- Invocar um pacote
- Remover um pacto

Pacotes PL/SQL: Visão Geral

- Pacotes PL/SQL:
  - Agrupa componentes logicamente relacionados:
    - Tipos PL/SQL
    - Variáveis, estruturas de dados e exceções
    - Subprogramas: procedimentos e funções
  - Consiste em duas partes:
    - Uma especificação
    - Um corpo

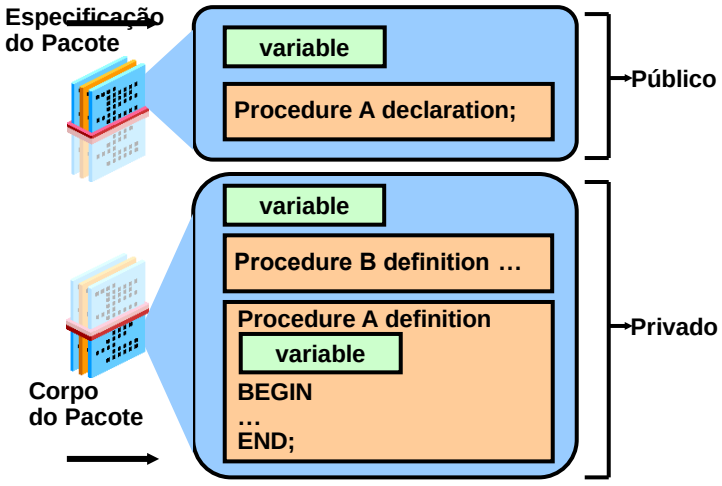
Pacotes PL / SQL: Visão geral

Os pacotes PL/SQL permitem agrupar tipos PL/SQL relacionados, variáveis, estruturas de dados, exceções e subprogramas em um único contêiner. Por exemplo, um pacote de recursos humanos pode conter procedimentos de contratação e demissão, funções de comissão e bônus e variáveis de isenção de impostos.

Um pacote consiste em duas partes armazenadas separadamente no banco de dados:

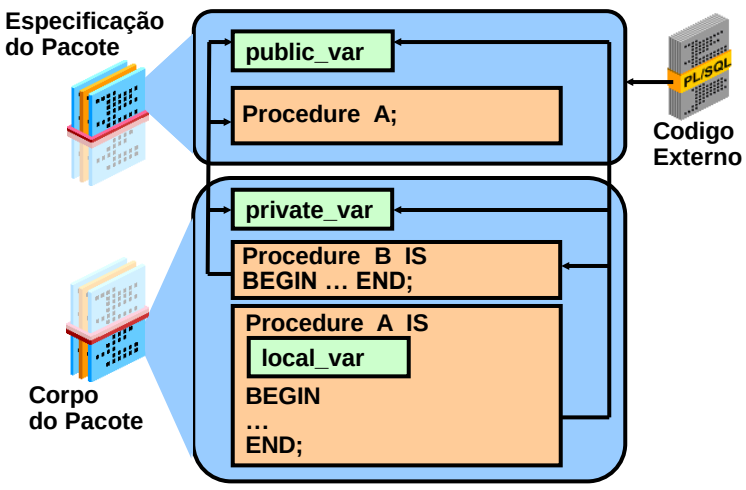
- Uma especificação
- Um corpo (opcional)

Componentes de um Pacote PL/SQL



Componentes de um pacote PL / SQL
Um pacote é criado em duas partes:
A especificação do pacote é a interface para seus aplicativos. Ele declara os tipos públicos, variáveis, constantes, exceções, cursores e subprogramas disponíveis para uso.
O corpo do pacote define seus próprios subprogramas e deve implementar completamente subprogramas declarados na parte de especificação. O corpo do pacote também pode definir construções PL/SQL, como variáveis de tipos, constantes, exceções e cursores.
Od componentes públicos podem ser referenciados a partir de qualquer ambiente de servidor Oracle externo ao pacote.
Os componentes privados são colocados no corpo do pacote e podem ser referenciados apenas por outras construções dentro do mesmo corpo. Os componentes privados podem fazer referência aos componentes públicos do pacote.

Visibilidade dos Componentes de um Pacote



Visibilidade dos componentes do pacote

A visibilidade de um componente descreve se esse componente pode ser visto, ou seja, referenciado e usado por outros componentes ou objetos. A visibilidade dos componentes depende se eles são declarados local ou globalmente.

Os componentes locais são visíveis dentro da estrutura em que são declarados:

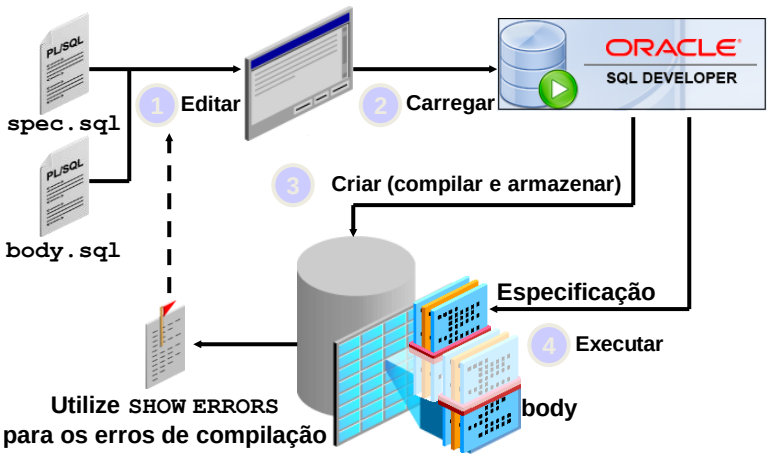
As variáveis definidas em um subprograma podem ser referenciadas dentro desse subprograma e não são visíveis para componentes externos - por exemplo, local_var pode ser usado no procedimento A.

As variáveis de pacote privado, que são declaradas em um corpo de pacote, podem ser referenciadas por outros componentes no mesmo corpo do pacote. Elas não são visíveis por nenhum subprograma ou objeto que esteja fora do pacote. Por exemplo, private_var pode ser usado pelos procedimentos A e B dentro do corpo do pacote, mas não fora. Os componentes declarados globalmente são visíveis internamente e externamente ao pacote, tais como:

Uma variável pública, que é declarada em uma especificação de pacote, pode ser referenciada e alterada fora do pacote (por exemplo, public_var pode ser referenciado externamente).

Um subprograma de pacote na especificação pode ser chamado a partir de fontes de código externas (por exemplo, o procedimento A pode ser chamado a partir de um ambiente externo ao pacote).

Desenvolvendo Pacotes PL/SQL



Desenvolvendo pacotes PL / SQL

Para desenvolver um pacote, execute as seguintes etapas:

1. Edite o texto para a especificação usando a instrução CREATE PACKAGE dentro de um arquivo de script SQL. Edite o texto para o corpo (se necessário, veja as diretrizes abaixo) usando a instrução CREATE PACKAGE BODY dentro de um arquivo de script SQL.
2. Carregue os arquivos de script em uma ferramenta, como o SQL Developer
3. Execute os arquivos de script para criar (ou seja, para compilar e armazenar) o pacote e o corpo do pacote.
4. Execute qualquer construção pública dentro da especificação do pacote.

Diretrizes para o desenvolvimento de pacotes

Considere salvar o texto para uma especificação de pacote e um corpo de pacote em dois arquivos diferentes para facilitar modificações.

Uma especificação de pacote pode existir sem um corpo de pacote; ou seja, quando a especificação do pacote não declara subprogramas, um corpo não é necessário. No entanto, um corpo de pacote não pode existir sem uma especificação de pacote.

Criando a Especificação do Pacote

- Sintaxe:

```
CREATE [OR REPLACE] PACKAGE package_name IS|AS
    public type and variable declarations
    subprogram specifications
END [package_name];
```

  - A opção OR REPLACE remove e recria a especificação do pacote.
  - As variáveis declaradas na especificação do pacote são NULL por padrão.
  - Todas as construções declaradas em uma especificação de pacote são visíveis para usuários que recebem privilégios no pacote.

Criando a Especificação do Pacote

Para criar pacotes, você declara todas as construções públicas dentro das especificações do pacote.

- Especifique a opção OU REPLACE, se desejar substituir uma especificação de pacote existente.
- Inicialize uma variável com um valor ou fórmula constante dentro da declaração, se necessário; Caso contrário, a variável é inicializada de forma implícita para NULL.

Sintaxe

- package_name especifica um nome para o pacote que deve ser exclusivo entre os objetos dentro do esquema de propriedade. Incluir o nome do pacote após a palavra-chave END é opcional.
- Declarações de tipo público e variável declara variáveis públicas, constantes, cursores, exceções, tipos definidos pelo usuário e subtipos.

Exemplo de Especificação do Pacote: comm_pkg

```
CREATE OR REPLACE PACKAGE comm_pkg IS
  std_comm NUMBER := 0.10;  --initialized to 0.10
  PROCEDURE reset_comm(new_comm NUMBER);
END comm_pkg;
/
```

- STD_COMM é uma variável global iniciada com 0.10.
- RESET_COMM é um procedimento público que redefine a comissão com base em algumas regras de negócios. É implementado no corpo do pacote.

Criando o Corpo do Pacote

- Sintaxe:

```
CREATE [OR REPLACE] PACKAGE BODY package_name IS|AS
    private type and variable declarations
    subprogram bodies
[BEGIN initialization statements]
END [package_name];
```

  - A opção OR REPLACE remove e recria o corpo do pacote.
  - Os identificadores definidos no corpo do pacote são privados e não visíveis fora do corpo do pacote.
  - Todas as construções privadas devem ser declaradas antes de serem referenciadas.
  - Construções públicas são visíveis nocorpo do pacote.

Criando o corpo do pacote

Crie um corpo de pacotes para definir e implementar todos os subprogramas públicos e suportar construções privadas. Ao criar um corpo de pacote, faça o seguinte:

- Especifique a opção OU REPLACE para substituir um corpo de pacote existente.
- Defina os subprogramas em uma ordem apropriada. O princípio básico é que você deve declarar uma variável ou subprograma antes que possa ser referenciado por outros componentes no mesmo corpo do pacote. É comum ver todas as variáveis privadas e subprogramas definidos primeiro e os subprogramas públicos definidos pelo último no corpo do pacote.
- O corpo do pacote deve completar a implementação para todos os procedimentos ou funções declarados nas especificações do pacote.

As seguintes são definições de itens na sintaxe do corpo do pacote:

- `package_name` especifica um nome para o pacote que deve ser o mesmo que a especificação do pacote. Usar o nome do pacote após a palavra-chave END é opcional.
- `private type and variable declarations` declares declaram variáveis privadas, constantes, cursores, exceções, tipos definidos pelo usuário e subtipos.
- a especificação do subprograma especifica a implementação completa de quaisquer procedimentos ou funções particulares e / ou públicas.
- `[BEGIN initialization statements]` é um bloco opcional de código de inicialização que é executado quando o pacote é referenciado pela primeira vez.

Exemplo de Corpo do Pacote: comm_pkg

```
CREATE OR REPLACE PACKAGE BODY comm_pkg IS
  FUNCTION validate(comm NUMBER) RETURN BOOLEAN IS
    max_comm employees.commission_pct%type;
  BEGIN
    SELECT MAX(commission_pct) INTO max_comm
    FROM   employees;
    RETURN (comm BETWEEN 0.0 AND max_comm);
  END validate;
  PROCEDURE reset_comm (new_comm NUMBER) IS BEGIN
    IF validate(new_comm) THEN
      std_comm := new_comm; -- reset public var
    ELSE  RAISE_APPLICATION_ERROR(
            -20230, 'Bad Commission');
    END IF;
  END reset_comm;
END comm_pkg;
```

Exemplo do corpo do pacote: comm_pkg

O slide mostra o corpo completo do pacote para comm_pkg, com uma função privada chamada validar para verificar uma comissão válida. A validação exige que a comissão seja positiva e maior do que a maior comissão existente. O procedimento reset_comm invoca a função de validação privada antes de alterar a comissão padrão. No exemplo, observe o seguinte:

- A variável std_comm referenciada no procedimento reset_comm é uma variável pública. As variáveis declaradas na especificação do pacote, como std_comm, podem ser direcionadas diretamente sem qualificação.
- O procedimento reset_comm implementa a definição pública na especificação.
- No corpo comm_pkg, a função de validação é privada e é direcionada diretamente do procedimento reset_comm sem qualificação.

Invocando Subprogramas no Pacote

- Chamando uma função dentro do mesmo pacote:

```
CREATE OR REPLACE PACKAGE BODY comm_pkg IS ...
  PROCEDURE reset_comm(new_comm NUMBER) IS
  BEGIN
    IF validate(new_comm) THEN
      std_comm := new_comm;
    ELSE ...
    END IF;
  END reset_comm;
END comm_pkg;
```

- Chamando um procedimento do pacote

```
EXECUTE comm_pkg.reset_comm(0.15)
```

- Chamando um procedimento de pacote em um esquema diferente:

```
EXECUTE scott.comm_pkg.reset_comm(0.15)
```

Invocando Subprogramas de Pacotes

Depois que o pacote é armazenado no banco de dados, você pode invocar subprogramas públicos ou privados dentro do mesmo pacote, ou subprogramas públicos, se externos ao pacote. Qualifique o subprograma com o nome do pacote quando invocado externamente ao pacote.

Removendo Pacotes

- Para remover a especificação do pacote e o corpo, use a seguinte sintaxe:

```
DROP PACKAGE package_name;
```

- Para remover o corpo do pacote, use a seguinte sintaxe:

```
DROP PACKAGE BODY package_name;
```

Removendo pacotes
Quando um pacote não é mais necessário, você pode usar uma instrução SQL DROP para removê-lo. Um pacote tem duas partes; portanto, você pode remover todo o pacote, ou você pode remover apenas o corpo do pacote e preservar a sua especificação .

Resumo

Comando	Ação
CREATE [OR REPLACE] PACKAGE	Criar [ou modificar] uma especificação de pacote existente
CREATE [OR REPLACE] PACKAGE BODY	Criar [ou modificar] um pacote existente
DROP PACKAGE	Remova a especificação do pacote e o corpo do pacote
DROP PACKAGE BODY	Remova somente o corpo do pacote

Summary (continued)

You can create, delete, and modify packages. You can remove both package specification and body by using the `DROP PACKAGE` command. You can drop the package body without affecting its specification.

OBRIGADO

profalexandre.barcelos@fiap.com.br

https://www.linkedin.com/in/alexandrebarcelos

FIAP