PACKAGE

1. Khái niệm:

Package là một tập hợp các kiểu dữ liệu, biến lưu giữ giá trị và các thủ tục, hàm có cùng một mối liên hệ với nhau, được gộp chung lại. Đặc điểm nổi bật nhất của package là khi một phần tử trong package được gọi thì toàn bộ nội dung của package sẽ được nạp vào trong hệ thống. Do đó, việc gọi tới các phần tử khác trong package sau này sẽ không phải mất thời gian nạp vào hệ thống nữa. Từ đó, nâng cao tốc độ thực hiện lệnh của toàn bộ hàm, thủ tục có trong package.

1. Lợi ích khi sử dụng package:
2. Tăng performance

Ngay khi gọi một hàm hay thủ tục bất kỳ trong package lần đầu tiên. Toàn bộ nội dung của package sẽ được nạp vào bộ nhớ. Do vậy, các hàm và thủ tục con trong package gọi đến sau này có thể thực hiện ngay mà không cần phải nạp lại vào bộ nhớ. Việc này làm giảm thiểu thao tác truy xuất vào ra (I/O access) nâng cao tốc độ.

Package ngăn việc biên dịch lại không cần thiết. Ví dụ, khi mình thay đổi thân của 1 package function thì Oracle Database sẽ không biên dịch lại các chương trình con gọi function đó, bởi vì các chương trình con chỉ quan tâm đến tham số và giá trị trả về đã được khai báo ở phần package spec

1. Dễ quản lý các store, function,… hơn.

Khi gom các store, function, kiểu dữ liệu, biến lưu giữ giá trị có liên quan với nhau lại thành 1 nhóm thì chúng ta sẽ dễ quản lý, dễ tìm khi cần hơn. Sau này nếu như có nhu cầu sửa đổi lại thì cũng dễ dàng hơn, đồng thời không phải nhìn cái list hàm, thủ tục dài quá trời quá đất.

Ví dụ như khi mình khai báo 1 biến để trong package thì biến này có thể sử dụng ở bất kì store procedures hay functions nào trong package ấy mà không phải khai báo lại.

1. Tăng tính phân nhỏ các thành phần (Modularity)

Ta có thể đóng gói các thành phần, cấu trúc có quan hệ logic với nhau trong cùng một module ứng với một package. Việc kế thừa giữa các package rất đơn giản, và được thực hiện một cách trong sáng.

1. Đơn giản trong việc thiết kế ứng dụng

Tất cả các thông tin cần thiết cho việc giao tiếp đều được đặt trong phần đặc tả của package (package specification). Nội dung phần này có thể được soạn thảo và biên dịch độc lập với phần thân của package (package body). Do đó, các hàm hay thủ tục có gọi tới các thành phần của package có thể được biên dịch tốt. Phần thân của package có thể được tiếp tục phát triển cho đến khi hoàn thành ứng dụng.

1. Ẩn dấu thông tin (hiding information)

Package cho phép sử dụng các thành phần bên trong dưới dạng public hay private. Tuỳ theo yêu cầu thiết kế, ta có thể cho phép truy nhập hay ẩn dấu thông tin. Từ đó, có thể bảo vệ được tính toàn vẹn dữ liệu.

1. Thực hiện quá tải (overloading)

Package cho phép thực hiện quá tải đối với các hàm và thủ tục trong nó. Theo đó, các hàm và thủ tục khác nhau có thể được phép đặt trùng tên. Việc này sẽ nâng cao tính mềm dẻo của việc sử dụng hàm, thủ tục trong package

1. Cấp quyền nhanh chóng

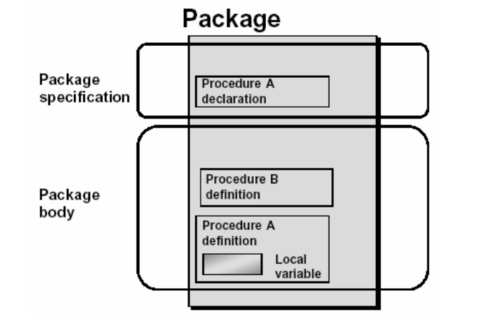
Thay vì phải cấp quyền lên từng đối tượng thì mình sẽ cấp quyền lên package, khi đó, các đối tượng trong package đó cũng bị ảnh hưởng theo.

1. Tình huống sử dụng:

Khi ứng dụng của mình quá phức tạp, quá nhiều hàm, thủ tục, biến. Như vậy sẽ rất khó quản lý cũng như hiệu suất làm việc, thực thi sẽ chậm đáng kể nếu như không gom nhóm các thành phần liên quan với nhau lại. Khi đó ta có thể sử dụng package để năng cao hiệu suất làm việc hơn.

1. Cấu trúc của package

Package được cấu trúc thành 2 phần chính: phần mô tả (specification) và phần thân (body). Phần mô tả khai báo các giao tiếp có thể có của package với bên ngoài. Phần thân là cài đặt các giao tiếp mà được khai báo ở phần mô tả trên.



1. Tạo package
2. Cú pháp:

|  |  |
| --- | --- |
|  | **-- Khai báo Package Spec:**    CREATE [OR REPLACE] PACKAGE <package\_name>  IS| AS  -- Phần khai báo các biến, hằng, cursor, ngoại lệ và  kiểu dữ liệu sử dụng trong toàn bộ package (public)  -- Khai báo đặc điểm các hàm thủ tục (public) bao gồm  các tham số đầu vào và các giá trị trả ra.  END <package\_name>;    **-- Khai báo phần Package Body:**  CREATE [OR REPLACE] PACKAGE BODY <package\_name>   IS | AS          -- Khai báo các kiểu chỉ sử dụng riêng trong package        -- Triển khai nội dung của các hàm, thủ tục khai báo trong Package Spec    END <package\_name>; |

1. Ví dụ

**Package Spec:**

Create Or Replace Package **Pkg\_Emp** Is

 -- Hàm trả về First\_Name

 Function Get\_First\_Name(p\_Emp\_Id Employee.Emp\_Id%Type)

    Return Employee.First\_Name%Type;

 -- Hàm trả về tên phòng ban của nhân viên.

 Function Get\_Dept\_Name(p\_Emp\_Id Employee.Emp\_Id%Type)

    Return Department.Name%Type;

End **Pkg\_Emp**;

**Package Body:**

Create Or Replace Package **Body** **Pkg\_Emp** Is

=====================================================

 Procedure Get\_Emp\_Infos(p\_Emp\_Id     Employee.Emp\_Id%Type

                        ,v\_First\_Name Out Employee.Emp\_Id%Type

                        ,v\_Last\_Name  Out Employee.Last\_Name%Type) As

 Begin

    …

 End;

 =============

 Function **Get\_First\_Name**(p\_Emp\_Id Employee.Emp\_Id%Type)

    Return Employee.First\_Name%Type As

    -- Khai báo một biến.

    v\_First\_Name Employee.First\_Name%Type;

    v\_Last\_Name  Employee.Last\_Name%Type;

 Begin

    -- Gọi sử dụng thủ tục Get\_Emp\_Infos

    Get\_Emp\_Infos(p\_Emp\_Id

                 ,v\_First\_Name -- Out

                 ,v\_Last\_Name -- Out

                  );

    --

    Return v\_First\_Name;

 End;

=============

 Function **Get\_Dept\_Name**(p\_Emp\_Id Employee.Emp\_Id%Type)

    Return Department.Name%Type As

    -- Khai báo một biến.

    v\_Dept\_Name Department.Name%Type;

 Begin

    Begin

…

…

    End;

    --

    Return v\_Dept\_Name;

 End;

End Pkg\_Emp;

**STANDARD** **Package bao** gồm các type, exceptions, và subprograms cơ bản đã được định nghĩa sẵn trong mỗi PL/SQL programs. Danh sách các type, exceptions, subprograms có trong standard package xem tại đây <http://psoug.org/reference/standard.html>

VD: Định nghĩa LIKE

function 'LIKE' (str VARCHAR2, pat VARCHAR2) return BOOLEAN;

function 'LIKE' (str varchar2, pat varchar2) return boolean is

begin

return peslik(str, pat);

end;

function peslik(str varchar2, pat varchar2) return boolean;

pragma interface (c,peslik);

Nguồn:

<http://docstore.mik.ua/orelly/oracle/bipack/ch01_04.htm>

<http://psoug.org/reference/standard.html>

<http://o7planning.org/web/fe/default/vi/document/179932/huong-dan-lap-trinh-oracle-pl-sql>

<https://docs.oracle.com/database/121/LNPLS/packages.htm#LNPLS009>

Và các file của thầy gởi.