PL/SQL Unit - 5

Page No.
Date: / /

Using PL/SQL Packages

What is a Package?

⇒ A pakage is an Oracle object which holds other objects within it. Objects commonly held within a package are procedures, functions, cursors, exceptions variables and constant. It is a way of creating generic, encapsulated reusable part.
A package once returned and debugged stored in Oracle system table and all the users who have the execute permission can execute the package.

X3 ## Advantages of Packages:-

- Packages enable the organization of commercial applications into efficient modules. Each package is easily understandable and the interfaces between packages are simple, clear and well-defined
- Packages allow granting of different privilages to different user groups more efficiently
- A package's public variables and cursors are available to the entire package and can be shared by all the elements of the package. This concept utilizes the memory more efficiently.

- Packages enable the overloading of procedures and function if necessary.
- Packages improve performance of the system by loading multiple objects into memory at once
- Packages promote the code reuseability. Therefore reducing redundant coding.
- Using packages we can create library of procedures functions and cursors that can be called from anywhere.

Components of Package :-

An oracle package normally consist of two components as package specification and package body.

(1) Package-Specification :
A package specification declares all the objects used within a package It mainly includes variables of record types, memory variables, constants, exception, cursors, function and procedure that are available for use.

The package specification contains:
1.) Name of the package.
2.) Name of the datatypes of arguement.
3. The declaration of all procedure, function etc

Mallab

Page No. 2
Date: / /

which are accessable from anywhere in the package.

Syntax:-

```
CREATE PACKAGE packageName
IS
    declarations:
      ----- -.
      -- - - ..
END packageName;
```

for eg:-
A simple package declaration consisting of a function and procedure will be as:-

```
CREATE PACKAGE emp-pack
IS
    FUNCTION Check_emp(eno IN NUMBER)
                        RETURN NUMBER;
    PROCEDURE inc_sal(eno IN NUMBER post
                        IN VARCHAR);
    END emp-pack;
```

Page No.
Date: / /

Package Body :-

The body of the package contains the definitions of public objects that have been declared in the package specification. The body can also contain other objects declarations that are private to the package. The object declared privately in the package body are not accessable to other object outside of the package.
The package body also contains definition of the subprogram.

After a package is returned, debugged compiled and stored in database application, users with permissions can call its sub-programs, use its cursors or raise its exceptions. A package body is created as :

```
CREATE OR REPLACE PACKAGE BODY
                          packageName
IS
    declarations;
    -------
    definitions;
    --------
```

For eg:

```
CREATE OR REPLACE PACKAGE BODY
                                emp_pack
IS
 FUNCTION check_emp (eno IN NUMBER)
                          RETURN NUMBER
  IS
      body of the function;
  END;
  PROCEDURE inc_sal (eno IN NUMBER,
                    post IN VARCHAR)
  IS
      body of the procedure;
  END;
END emp_pack;
```

Public & Private Objects:-

The sub-programs, cursors, variables and constants declared inside the package body are private and such private objects are only accessible within the same package and cannot be accessed in PL/SQL block outside the package. Such private objects are used for internal calculation

purpose of any package.
Any subprograms, cursors, variables and constants declared within package specifications are the public objects. Public objects become visible inside the same package. as well as outside the package. So public objects are shareable objects and they optimize the use of memory. To access public-objects outside the package we use:

packageName . objectName ~~statement~~

PL/ SQL

Overloading Procedures & functions.

Page No.
Date: / /

Overloading procedures & functions:-

In Oracle package we can declare and define more than 1 function or procedure with the same name but with different parameters interms of either number of parameters or their data types. This concept is known as overloading functions or procedures. This overloadin g approach simplifies the PL/SQL code block by providing the similar name to a class of procedures or functions that can perform similar types of operations but on different parameters.
For example:-

Create a package having two functions, one to check that the numerical value is greater than zero and other function to check that date is less than or equal to system date using overloaded functions.

→

Page No.
Date: / /

```
CREATE OR REPLACE PACKAGE check_fun
IS
   FUNCTION value_ok(num IN NUMBER)
                        RETURN VARCHAR2;
   FUNCTION value_ok (dt IN DATE) RETURN
                        VARCHAR2;

 END;
 CREATE OR REPLACE PACKAGE BODY check_fun
 IS
    FUNCTION value_ok (num IN NUMBER)
                        RETURN VARCHAR2

      IS
        IF num >0 THEN
            RETURN 'True Result of overloaded pro
                        function 1';
      ELSE
          RETURN 'False Result of overloaded
                        function 1';
      END IF;
  END;
FUNCTION value_ok (dt IN DATE) RETURN
                        VARCHAR2
  IS
     IF dt<= SYSDATE THEN
        RETURN 'True Result of overloaded
                   function 2';
```

```
ELSE
    RETURN 'False Result of overloaded function2';
  ENDIF;
END;
END check_fun;
```

* What are the advantages of overloading sub-programs?

* Considering a database having Emp & Dept Tables. Define a package having one ~~function~~ procedure that takes employee number as input and increase the salary of that employee by 10%. Another procedure ~~function~~ that takes post as input and increase the salary of all employees of that post by 5%. A procedure that takes post as input and displays the total number of employees in that post.

PL/SQL

Product Specific Packages:

Oracle and various oracle tools are supplied with product specific packages that define API's and we can call them from PL/SQL, Java and other programming environment. Few commonly used packages are;

(a) DBMS_ALERT :-

This package allows database triggers and application programs to display certain types of errors or warning messages when transactions are executed and they perform some WRITE operations on database. The alerts are transaction based and they operate independently.

(b) DBMS_OUTPUT :-

This package allows us to display output from PL/SQL programs or subprograms ~~[illegible]~~ which makes it easier to debug and test the program. The method put_line ( ) which is used to display some message on output screen.

Page No.
Date: / /

(c) ~~DBMS_PIPE:-~~

(c) DBMS_PIPE :

A pipe is an area of memory used by one process to pass information to other process. This package allows different sessions to communicate over different pipes. This package is useful in various operating systems to interact with high-level language programs.

(d) UTL_FILE:-

This package allows PL/SQL programs to read and write operating system text files. It provides different file related operations like OPEN( ), GET( ), PUT( ), CLOSE( ) etc. to interact with files for various reading and writing operations.

(e) UTL_HTTP:-

This package allows PL/SQL programs to make hyper text transfer protocol callouts that means it can retrieve data from the internet or ORACLE web server as well as can provide data to them. It exchanges data in the form of HTML.

Page No.
Date: / /

(f.) UTL_SMTP:
This package allows PL/SQL program to interact with email servers for various email related operations.

(g.) HTF :

(h.) HTP :

Scanned with CamScanner