Creating Packages

What Are PL/SQL Packages?

- A package is a schema object that groups logically related PL/SQL types, variables, and subprograms.
- Packages usually have two parts:
  - A specification (spec)
  - A body
- The specification is the interface to the package. It declares the types, variables, constants, exceptions, cursors, and subprograms that can be referenced from outside the package.
- The body defines the queries for the cursors and the code for the subprograms.
- Enable the Oracle server to read multiple objects into memory at once.

Advantages of Using Packages

- Modularity: Encapsulating related constructs
- Easier maintenance: Keeping logically related functionality together
- Easier application design: Coding and compiling the specification and body separately
- Hiding information:
  - Only the declarations in the package specification are visible and accessible to applications
  - Private constructs in the package body are hidden and inaccessible
  - All coding is hidden in the package body

Creating the Package Specification: Using the CREATE PACKAGE Statement

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

- The OR REPLACE option drops and re-creates the package specification.
- Variables declared in the package specification are initialized to NULL by default.
- All the constructs declared in a package specification are visible to users who are granted privileges on the package.

Creating the Package Body

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

- The `OR REPLACE` option drops and re-creates the package body.
- Identifiers defined in the package body are *private* and not visible outside the package body.
- All *private* constructs must be declared before they are referenced.
- Public constructs are visible to the package body.

Guidelines for Writing Packages

- Develop packages for general use.
- Define the package specification before the body.
- The package specification should contain only those constructs that you want to be public.
- Place items in the declaration part of the package body when you must maintain them throughout a session or across transactions.
- The fine-grain dependency management reduces the need to recompile referencing subprograms when a package specification changes.
- The package specification should contain as few constructs as possible.

Guidelines for Writing Packages

The package specification should contain as few constructs as possible.

Package Specification

Procedure insert emp
Procedure update emp_sal
Procedure delete emp

Try to keep package specification Simple as you can

Package body

Proceure check_before_insert
Proceure check_before_delete
Procedure insert emp
Procedure update emp_sal
Procedure delete emp

Try to add additional codes you need in package body as private subprograms

The Visibility of a Package's Components

```
create or replace package p_test
is
c_var1 constant number:=10;
c_var2 varchar2(100):='welcome';

procedure print;
end;
```

c_var1/c_var2 can be referenced any place in package body

c_var3 can be referenced any place in package body

c_var4 can be referenced only in print procedure

```
create or replace package body p_test
is
c_var3 varchar2(100):='hi there';
 procedure print
 is
 c_var4 varchar2(100):='hi';
 begin
 dbms_output.put_line('this variable came from package spec. '||c_var1);
 dbms_output.put_line('this variable came from package spec. '||c_var2);
 dbms_output.put_line('this variable came from package body. '||c_var3);
  dbms_output.put_line('this variable came from print Proc. '||c_var4);
 end;

end;

execute p_test.print;
```

Dictionary tables and Dropping the Package

```
select * from user_objects
where oBject_name='P_TEST'

SELECT * FROM USER_SOURCE
WHERE NAME='P_TEST'
AND TYPE='PACKAGE';

SELECT * FROM USER_SOURCE
WHERE NAME='P_TEST'
AND TYPE='PACKAGE BODY';

--to drop package specification and body

drop package p_test;

--to drop only package body

drop package body p_test;
```

Thank You