

PL/SQL Packages:

- Bundle of things
- Group of logically related sub programmes
- It can be Procedures, functions or cursors

Packages has two parts: → These parts can be compiled separately

1. Specification
 - a. It declares types, variables, constants, exceptions, cursors and subprogrammes that can be referenced from outside the packages.
 - b. It's an interface of the packages
2. Body
 - a. Cannot be compiled if we don't have specification
 - b. Defines queries for the cursors and the code for the subprogrammes
 - c. We can use two types of visibility: Private or Local

Persistent Object:

- Invoked Once, used many times for a user context because it is saved in memory → known as UGA (User Global Area)
- SGA (System Global Area) will be deallocated when system is shutdown
- It is good that all the object is saved in SGA or PGA or UGA because we don't have to access disk again and again

Packages Benefits (Advantages):

1. Persistent → Invoked Once and Reuse again during current session (Reusability of Objects)
 - It gives better performance because when it is called first it will be loaded in memory and after that it calls related subprogrammes in the packages
 - It reduces disk I/O.
2. Declaration in the specification are PUBLIC so it can be accessed easily
3. Data Encapsulations: Implementation in package Body is hidden (Private/Local)
4. Overloading: Multiple Program of Same name

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.
- When we have only constant then at that time just use Package Specification.

Access Control:

- Privileges: Assigning privileges to user
- Roles: defining roles for user

Notes:

- Procedure inside the package can only be used with dot operator.
 - For Example, comm_pkg.reset_comm(0.10);

Controlling Side-Effects of PL/SQL Programs:

- Mutation:
 - When called from a SELECT or parallelized DML statements, function cannot modify that particular table or modify. Also known as **concurrency control problem**
 - Dead Lock situation. → Need to use rollback, commit, on kill session
- When called from a DML Statement, function cannot query or modify any database tables modified by the statement