Un **paquete** en PL/SQL es una estructura que agrupa y organiza procedimientos, funciones, tipos de datos, excepciones, cursores y otras declaraciones, con el objetivo de facilitar la gestión y reutilización del código. Se divide en dos partes principales: la **cabecera** y el **cuerpo**.

**Cabecera del paquete:**

La cabecera contiene las **declaraciones públicas**, es decir, los elementos del paquete que serán accesibles desde cualquier parte de la aplicación. Aquí se definen los tipos de datos, constantes, variables, subprogramas (procedimientos y funciones) y cursores que se podrán utilizar fuera del paquete.

La sintaxis para crear la cabecera es:

CREATE [OR REPLACE] PACKAGE nombre\_paquete AS

-- Declaraciones públicas

<especificación de subprogramas>

END nombre\_paquete;

**Cuerpo del paquete:**

El cuerpo contiene la **implementación** de los subprogramas (procedimientos, funciones) y las **declaraciones privadas**, es decir, los detalles internos del paquete que solo pueden ser accedidos dentro de él. La cabecera y el cuerpo se pueden compilar independientemente.

La sintaxis para crear el cuerpo es:

CREATE [OR REPLACE] PACKAGE BODY nombre\_paquete AS

-- Declaraciones privadas

<código de subprogramas>

[BEGIN

-- Instrucciones iniciales]

END nombre\_paquete;

**Ejemplo de paquete:**

**Cabecera:**

CREATE OR REPLACE PACKAGE buscar\_emple AS

TYPE t\_reg\_emple IS RECORD (

num\_empleado emple.emp\_no%TYPE,

apellido emple.apellido%TYPE,

salario emple.salario%TYPE,

departamento emple.dept\_no%TYPE

);

PROCEDURE ver\_por\_numero(v\_cod emple.emp\_no%TYPE);

PROCEDURE ver\_por\_apellido(v\_ape emple.apellido%TYPE);

FUNCTION datos(v\_cod emple.emp\_no%TYPE) RETURN t\_reg\_emple;

END buscar\_emple;

**Cuerpo:**

CREATE OR REPLACE PACKAGE BODY buscar\_emple AS

vg\_emple t\_reg\_emple;

PROCEDURE ver\_emple IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE(vg\_emple.num\_empleado || '\*' || vg\_emple.apellido || '\*' || vg\_emple.salario || '\*' || vg\_emple.departamento);

END ver\_emple;

PROCEDURE ver\_por\_numero(v\_cod emple.emp\_no%TYPE) IS

BEGIN

SELECT emp\_no, apellido, salario, dept\_no INTO vg\_emple FROM emple WHERE emp\_no = v\_cod;

ver\_emple;

END ver\_por\_numero;

PROCEDURE ver\_por\_apellido(v\_ape emple.apellido%TYPE) IS

BEGIN

SELECT emp\_no, apellido, salario, dept\_no INTO vg\_emple FROM emple WHERE apellido = v\_ape;

ver\_emple;

END ver\_por\_apellido;

FUNCTION datos(v\_cod emple.emp\_no%TYPE) RETURN t\_reg\_emple IS

BEGIN

SELECT emp\_no, apellido, salario, dept\_no INTO vg\_emple FROM emple WHERE emp\_no = v\_cod;

RETURN vg\_emple;

END datos;

END buscar\_emple;

**Uso de los objetos definidos en los paquetes:**

1. **Desde dentro del paquete**: Los objetos definidos en la cabecera del paquete pueden ser utilizados dentro del cuerpo del mismo. No es necesario el prefijo execute, ya que están en el mismo contexto.

Ejemplo:

v\_emple := buscar\_emple.datos(v\_n\_ape);

1. **Desde fuera del paquete**: Para utilizar los objetos definidos en un paquete desde fuera, deben haber sido declarados en la cabecera. La llamada se hace mediante nombre\_paquete.nombre\_objeto.

Ejemplo:

EXECUTE buscar\_emple.ver\_por\_numero(123);

**Declaración de cursores en paquetes:**

También podemos declarar **cursores** dentro de los paquetes. Los cursores son objetos que permiten trabajar con resultados de consultas SQL. La declaración del cursor se realiza en la cabecera del paquete y su implementación en el cuerpo.

Ejemplo de cursor en paquete:

CREATE OR REPLACE PACKAGE empleados AS

CURSOR a RETURN empleado%ROWTYPE;

END empleados;

CREATE OR REPLACE PACKAGE BODY empleados AS

CURSOR a RETURN empleado%ROWTYPE IS

SELECT \* FROM empleado WHERE salario < 10000;

END empleados;

Este ejemplo define un cursor llamado a que selecciona todos los empleados con salario inferior a 10,000.