¡Claro! Ahora te mostraré cómo trabajar con **cursos** dentro de un paquete utilizando **RECORD** y **ROWTYPE** en PL/SQL.

**1. Ejemplo con RECORD y cursos en un paquete**

**Paso 1: Crear el paquete y la especificación**

En la especificación del paquete, declaramos un procedimiento que usará un **cursor** para obtener los datos de los empleados.

CREATE OR REPLACE PACKAGE emp\_package IS

-- Declaramos el procedimiento que usará un cursor con RECORD

PROCEDURE mostrar\_empleados\_record;

END emp\_package;

/

**Paso 2: Crear el cuerpo del paquete**

Aquí en el cuerpo del paquete, implementamos el procedimiento utilizando un **cursor** explícito y **RECORD**.

CREATE OR REPLACE PACKAGE BODY emp\_package IS

-- Definir un tipo RECORD con las columnas de EMP

TYPE emp\_record IS RECORD (

empno EMP.EMPNO%TYPE,

ename EMP.ENAME%TYPE

);

-- Crear un cursor explícito

CURSOR emp\_cursor IS

SELECT EMPNO, ENAME

FROM EMP;

-- Implementación del procedimiento que usa el cursor con RECORD

PROCEDURE mostrar\_empleados\_record IS

emp\_rec emp\_record; -- Variable para almacenar los datos de un solo registro

BEGIN

-- Abrir el cursor

OPEN emp\_cursor;

-- Bucle para recorrer los registros obtenidos del cursor

LOOP

FETCH emp\_cursor INTO emp\_rec; -- Cargar cada fila en emp\_rec

EXIT WHEN emp\_cursor%NOTFOUND; -- Salir del bucle cuando no haya más registros

-- Imprimir el número de empleado y el nombre

DBMS\_OUTPUT.PUT\_LINE('Empleado: ' || emp\_rec.empno || ', Nombre: ' || emp\_rec.ename);

END LOOP;

-- Cerrar el cursor

CLOSE emp\_cursor;

END mostrar\_empleados\_record;

END emp\_package;

/

**2. Ejemplo con ROWTYPE y cursos en un paquete**

**Paso 1: Crear el paquete y la especificación**

En la especificación del paquete, declaramos un procedimiento que usará un **cursor** con **ROWTYPE**.

CREATE OR REPLACE PACKAGE emp\_package IS

-- Declaramos el procedimiento que usará un cursor con ROWTYPE

PROCEDURE mostrar\_empleados\_rowtype;

END emp\_package;

/

**Paso 2: Crear el cuerpo del paquete**

En el cuerpo del paquete, implementamos el procedimiento utilizando un **cursor** explícito con **ROWTYPE**.

CREATE OR REPLACE PACKAGE BODY emp\_package IS

-- Crear un cursor explícito usando ROWTYPE para la tabla EMP

CURSOR emp\_cursor IS

SELECT \*

FROM EMP;

-- Implementación del procedimiento que usa el cursor con ROWTYPE

PROCEDURE mostrar\_empleados\_rowtype IS

emp\_rec EMP%ROWTYPE; -- Variable para almacenar los datos de un solo registro

BEGIN

-- Abrir el cursor

OPEN emp\_cursor;

-- Bucle para recorrer los registros obtenidos del cursor

LOOP

FETCH emp\_cursor INTO emp\_rec; -- Cargar cada fila en emp\_rec

EXIT WHEN emp\_cursor%NOTFOUND; -- Salir del bucle cuando no haya más registros

-- Imprimir el número de empleado y el nombre

DBMS\_OUTPUT.PUT\_LINE('Empleado: ' || emp\_rec.empno || ', Nombre: ' || emp\_rec.ename);

END LOOP;

-- Cerrar el cursor

CLOSE emp\_cursor;

END mostrar\_empleados\_rowtype;

END emp\_package;

/

**3. Llamar a los procedimientos desde un bloque anónimo**

Ahora puedes llamar a los procedimientos de este paquete desde un bloque anónimo o cualquier otra parte de tu aplicación PL/SQL.

**Llamada al procedimiento con RECORD:**

BEGIN

-- Llamar al procedimiento que usa RECORD y el cursor explícito

emp\_package.mostrar\_empleados\_record;

END;

/

**Llamada al procedimiento con ROWTYPE:**

BEGIN

-- Llamar al procedimiento que usa ROWTYPE y el cursor explícito

emp\_package.mostrar\_empleados\_rowtype;

END;

/

**Explicación:**

1. **Definición de tipos**:
   * Con **RECORD**, creamos un tipo de registro (emp\_record) que solo contiene las columnas que necesitamos (en este caso, empno y ename).
   * Con **ROWTYPE**, la variable emp\_rec tiene el mismo tipo que una fila completa de la tabla EMP, lo que facilita trabajar con toda la estructura de la tabla.
2. **Cursor explícito**:
   * En ambos ejemplos, definimos un **cursor explícito** (emp\_cursor) para seleccionar datos de la tabla EMP.
   * Luego, utilizamos un **bucle LOOP** con FETCH para recorrer cada registro obtenido por el cursor, cargándolos en la variable emp\_rec.
3. **Manejo del cursor**:
   * Usamos OPEN para abrir el cursor, FETCH para obtener cada registro y CLOSE para cerrarlo cuando hemos terminado de procesar los datos.
   * La condición de salida del bucle es emp\_cursor%NOTFOUND, que verifica si hemos llegado al final del conjunto de resultados.
4. **Impresión de resultados**:
   * Dentro del bucle, utilizamos DBMS\_OUTPUT.PUT\_LINE para imprimir los datos de cada empleado, que son almacenados en emp\_rec.

**Resumen:**

* Los **cursos** permiten recuperar filas de la base de datos de forma secuencial y almacenar cada fila en una variable de tipo **RECORD** o **ROWTYPE**.
* Puedes utilizar **cursos explícitos** con **RECORD** o **ROWTYPE** dentro de un **paquete** para iterar sobre los datos de la tabla y realizar operaciones (como imprimir los datos) de manera eficiente.
* Los procedimientos en el paquete gestionan el cursor, lo abren, iteran sobre los datos y finalmente lo cierran.

¡Con esto, tienes una idea completa de cómo usar **cursos** con **RECORD** y **ROWTYPE** dentro de un **paquete** en PL/SQL! Si tienes más dudas o necesitas más ejemplos, no dudes en preguntar.