¡Claro! Ahora te proporcionaré los mismos ejemplos pero dentro de un procedimiento en lugar de un bloque anónimo. Verás cómo se estructuran los procedimientos para utilizar **RECORD** y **ROWTYPE** en PL/SQL.

**Ejemplo 1: Usando RECORD dentro de un procedimiento**

Este ejemplo crea un procedimiento que utiliza un tipo **RECORD** y una colección de ese tipo para almacenar y procesar los datos de la tabla EMP.

CREATE OR REPLACE PROCEDURE procesar\_empleados\_record IS

-- Definir un tipo RECORD con columnas específicas de EMP

TYPE emp\_record IS RECORD (

empno EMP.EMPNO%TYPE,

ename EMP.ENAME%TYPE

);

-- Crear una colección de tipo RECORD

TYPE emp\_table IS TABLE OF emp\_record;

-- Variable para almacenar la colección de registros

emp\_collection emp\_table;

BEGIN

-- Usar BULK COLLECT para llenar la colección con datos de la tabla EMP

SELECT EMPNO, ENAME

BULK COLLECT INTO emp\_collection

FROM EMP;

-- Iterar sobre la colección de registros y mostrar los resultados

FOR i IN 1 .. emp\_collection.COUNT LOOP

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

END LOOP;

END procesar\_empleados\_record;

/

**Explicación:**

1. **Definición del RECORD**: Se define un tipo emp\_record con solo las columnas EMPNO y ENAME de la tabla EMP.
2. **Colección de tipo RECORD**: Se crea una colección emp\_collection de tipo emp\_table que almacenará registros de tipo emp\_record.
3. **Uso de BULK COLLECT**: Se utiliza para llenar la colección con los registros seleccionados de la tabla EMP.
4. **Iteración**: Se recorre la colección y se imprimen los valores de EMPNO y ENAME.

**Ejemplo 2: Usando ROWTYPE dentro de un procedimiento**

Este ejemplo utiliza **%ROWTYPE** para crear un procedimiento que devuelve todos los registros de la tabla EMP en una colección.

CREATE OR REPLACE PROCEDURE procesar\_empleados\_rowtype IS

-- Crear una colección de registros de tipo %ROWTYPE (es decir, una fila completa de EMP)

TYPE emp\_table IS TABLE OF EMP%ROWTYPE;

-- Variable para almacenar la colección de registros

emp\_collection emp\_table;

BEGIN

-- Usar BULK COLLECT para llenar la colección con datos de la tabla EMP

SELECT \*

BULK COLLECT INTO emp\_collection

FROM EMP;

-- Iterar sobre la colección de registros y mostrar los resultados

FOR i IN 1 .. emp\_collection.COUNT LOOP

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

END LOOP;

END procesar\_empleados\_rowtype;

/

**Explicación:**

1. **Uso de ROWTYPE**: Se define emp\_table como una colección de registros de tipo EMP%ROWTYPE, lo que significa que la colección tendrá todos los campos de la tabla EMP.
2. **Colección de tipo ROWTYPE**: La colección emp\_collection almacenará registros de tipo EMP%ROWTYPE.
3. **Uso de BULK COLLECT**: Se usa para llenar la colección con los registros de la tabla EMP.
4. **Iteración**: Al igual que en el primer ejemplo, se recorre la colección y se imprimen los valores de EMPNO y ENAME.

**Llamar a los Procedimientos**

Para llamar a los procedimientos que acabamos de crear, puedes ejecutar el siguiente código en un bloque anónimo:

BEGIN

-- Llamar al procedimiento con RECORD

procesar\_empleados\_record;

-- Llamar al procedimiento con ROWTYPE

procesar\_empleados\_rowtype;

END;

/

**Comparación entre ambos enfoques:**

* **RECORD** te permite definir un tipo de registro personalizado con solo las columnas que necesitas. Esto es útil si solo deseas trabajar con un subconjunto de columnas de la tabla.
* **ROWTYPE** crea un registro que refleja la estructura completa de la tabla. Esto es útil si deseas trabajar con todas las columnas de la tabla, sin tener que definirlas manualmente.

En resumen, si solo necesitas algunas columnas de una tabla, usa **RECORD**. Si necesitas trabajar con todos los campos de la tabla, usa **ROWTYPE**.

¡Espero que esto te haya quedado claro! Si necesitas más detalles o ajustes, no dudes en pedírmelo.