**Claro, te proporcionaré ejemplos utilizando procedimientos tanto para RECORD como para %ROWTYPE, donde la lógica de la colección de registros se maneja dentro del procedimiento. Los ejemplos serán similares al anterior, pero esta vez usaremos procedimientos en lugar de funciones.**

**Procedimiento usando RECORD:**

**Este procedimiento obtiene registros de la tabla EMP y los almacena en una colección de tipo RECORD:**

**DECLARE**

**-- Definir un tipo de RECORD que se ajusta a la estructura de la tabla EMP**

**TYPE EMP\_RECORD IS RECORD (**

**EMPNO EMP.EMPNO%TYPE,**

**ENAME EMP.ENAME%TYPE,**

**JOB EMP.JOB%TYPE,**

**MGR EMP.MGR%TYPE,**

**HIREDATE EMP.HIREDATE%TYPE,**

**SAL EMP.SAL%TYPE,**

**COMM EMP.COMM%TYPE,**

**DEPTNO EMP.DEPTNO%TYPE**

**);**

**-- Definir un tipo de colección para almacenar múltiples registros de tipo EMP\_RECORD**

**TYPE TABLA\_EMPLEADOS IS TABLE OF EMP\_RECORD;**

**-- Variable para almacenar la colección de registros**

**VAUX\_TABLA\_LLAMADA TABLA\_EMPLEADOS;**

**-- Procedimiento para obtener la colección de registros**

**PROCEDURE OBTENER\_EMPLEADOS IS**

**BEGIN**

**-- Realizar la consulta y almacenar los registros en la colección usando un bucle**

**FOR rec IN (SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO**

**FROM EMP) LOOP**

**-- Añadir los registros a la colección**

**VAUX\_TABLA\_LLAMADA.EXTEND;**

**VAUX\_TABLA\_LLAMADA(VAUX\_TABLA\_LLAMADA.COUNT) := EMP\_RECORD(rec.EMPNO, rec.ENAME, rec.JOB, rec.MGR, rec.HIREDATE, rec.SAL, rec.COMM, rec.DEPTNO);**

**END LOOP;**

**END OBTENER\_EMPLEADOS;**

**BEGIN**

**-- Llamar al procedimiento para obtener los empleados**

**OBTENER\_EMPLEADOS;**

**-- Iterar sobre la colección y mostrar los resultados**

**FOR i IN 1 .. VAUX\_TABLA\_LLAMADA.COUNT LOOP**

**DBMS\_OUTPUT.PUT\_LINE('Empleado: ' || VAUX\_TABLA\_LLAMADA(i).EMPNO || ', ' || VAUX\_TABLA\_LLAMADA(i).ENAME);**

**END LOOP;**

**END;**

**/**

**Explicación:**

1. **Definición de RECORD:**
   * **Se crea el tipo EMP\_RECORD que describe la estructura de cada registro de la tabla EMP.**
2. **Colección de RECORD:**
   * **Se define TABLA\_EMPLEADOS como una colección de registros de tipo EMP\_RECORD.**
3. **Procedimiento OBTENER\_EMPLEADOS:**
   * **Dentro del procedimiento, se realiza una consulta para obtener los registros de la tabla EMP y luego se añaden a la colección VAUX\_TABLA\_LLAMADA.**
4. **Iteración:**
   * **El bloque BEGIN llama al procedimiento OBTENER\_EMPLEADOS y luego recorre la colección mostrando los resultados con DBMS\_OUTPUT.PUT\_LINE.**

**Procedimiento usando %ROWTYPE:**

**Este procedimiento es similar, pero esta vez utilizamos %ROWTYPE para definir el tipo de cada registro, lo que lo hace más directo y dinámico en cuanto a las columnas de la tabla:**

**DECLARE**

**-- Definir un tipo de colección basado en %ROWTYPE de la tabla EMP**

**TYPE TABLA\_EMPLEADOS IS TABLE OF EMP%ROWTYPE;**

**-- Variable para almacenar la colección de registros**

**VAUX\_TABLA\_LLAMADA TABLA\_EMPLEADOS;**

**-- Procedimiento para obtener la colección de registros**

**PROCEDURE OBTENER\_EMPLEADOS IS**

**BEGIN**

**-- Realizar la consulta y almacenar los registros en la colección usando BULK COLLECT**

**SELECT \***

**BULK COLLECT INTO VAUX\_TABLA\_LLAMADA**

**FROM EMP;**

**END OBTENER\_EMPLEADOS;**

**BEGIN**

**-- Llamar al procedimiento para obtener los empleados**

**OBTENER\_EMPLEADOS;**

**-- Iterar sobre la colección y mostrar los resultados**

**FOR i IN 1 .. VAUX\_TABLA\_LLAMADA.COUNT LOOP**

**DBMS\_OUTPUT.PUT\_LINE('Empleado: ' || VAUX\_TABLA\_LLAMADA(i).EMPNO || ', ' || VAUX\_TABLA\_LLAMADA(i).ENAME);**

**END LOOP;**

**END;**

**/**

**Explicación:**

1. **Definición de colección de tipo %ROWTYPE:**
   * **Usamos EMP%ROWTYPE para definir la colección TABLA\_EMPLEADOS, lo que hace que cada elemento de la colección tenga la misma estructura que una fila de la tabla EMP.**
2. **Procedimiento OBTENER\_EMPLEADOS:**
   * **Este procedimiento realiza la consulta con BULK COLLECT y almacena los resultados directamente en la colección VAUX\_TABLA\_LLAMADA.**
3. **Iteración:**
   * **Después de llamar al procedimiento OBTENER\_EMPLEADOS, el bloque recorre la colección VAUX\_TABLA\_LLAMADA y muestra cada registro con DBMS\_OUTPUT.PUT\_LINE.**

**Resumen:**

* **RECORD te permite definir explícitamente la estructura de un solo registro, lo que te da más control sobre los tipos de datos de cada campo.**
* **%ROWTYPE se ajusta automáticamente a la estructura de una tabla, lo que hace que el código sea más flexible y se mantenga sincronizado con cualquier cambio en la tabla.**

**En ambos casos, el procedimiento encapsula la lógica de la consulta, mientras que el bloque anónimo se encarga de llamar al procedimiento y mostrar los resultados. ¡Espero que te haya quedado claro! Si tienes más dudas, no dudes en preguntar.**