**PASOS SI USO EL RECORD COMO ESTRUCURA**

**1. Crear un tipo de dato de registro**

Primero, debes definir un tipo de registro que refleje la estructura de los datos que deseas devolver.

-- Definir un tipo de registro para la tabla de registros DBE IR DENTOR DE UN BLOQUE PL/SQL FUERA NO FUNCIONAN

CREATE OR REPLACE TYPE registro\_tipo AS RECORD (

id NUMBER,

nombre VARCHAR2(100),

edad NUMBER

);

Aquí, hemos creado un tipo de objeto (registro\_tipo) que contiene tres campos: id, nombre, y edad. Puedes ajustarlo a la estructura que necesites.

**2. Crear un tipo de colección (tabla de registros)**

Luego, necesitas crear un tipo de colección que contenga múltiples registros de tipo registro\_tipo. Vamos a usar un tipo de tabla para esto.

-- Definir un tipo de tabla para almacenar múltiples registros

CREATE OR REPLACE TYPE tabla\_registros\_tipo AS TABLE OF registro\_tipo;

Este tipo de colección (tabla\_registros\_tipo) es una tabla de registros del tipo registro\_tipo, y se usará para devolver múltiples registros.

**3. Crear la función que devolverá la tabla de registros**

Ahora puedes crear una función que devolverá esta colección de registros. En este ejemplo, vamos a suponer que estamos seleccionando datos de una tabla llamada empleados y devolviendo estos registros.

-- Crear la función que devuelve una tabla de registros

CREATE OR REPLACE FUNCTION obtener\_empleados

RETURN tabla\_registros\_tipo

IS

empleados\_tabla tabla\_registros\_tipo; -- Variable para almacenar los registros

BEGIN

-- Seleccionamos los datos y los almacenamos en la colección

SELECT registro\_tipo(id, nombre, edad)

BULK COLLECT INTO empleados\_tabla

FROM empleados;

-- Devolver la tabla de registros

RETURN empleados\_tabla;

END;

En esta función:

* empleados\_tabla es una variable de tipo tabla\_registros\_tipo, donde almacenaremos los registros seleccionados.
* Usamos BULK COLLECT INTO para llenar la colección empleados\_tabla con los registros obtenidos de la tabla empleados.
* Finalmente, la función devuelve empleados\_tabla, que contiene la tabla de registros.

**4. Llamar a la función**

Ahora que tienes la función creada, puedes llamarla en un bloque PL/SQL para obtener los registros devueltos.

DECLARE

empleados tabla\_registros\_tipo; -- Variable para almacenar el resultado de la función

BEGIN

-- Llamar a la función y obtener los registros

empleados := obtener\_empleados;

-- Mostrar los resultados (por ejemplo, en un loop)

FOR i IN 1 .. empleados.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('ID: ' || empleados(i).id || ', Nombre: ' || empleados(i).nombre || ', Edad: ' || empleados(i).edad);

END LOOP;

END;

Este bloque:

* Llama a la función obtener\_empleados y almacena el resultado en la variable empleados.
* Luego, recorre la colección empleados y muestra cada registro usando DBMS\_OUTPUT.PUT\_LINE.

**Resumen**

PASO SI USO EL RECORD

1. Definimos un tipo de objeto (registro\_tipo) para estructurar los datos de cada registro.
2. Creamos un tipo de colección (tabla\_registros\_tipo) para almacenar múltiples registros.
3. Creamos una función (obtener\_empleados) que selecciona datos de una tabla y los devuelve como una colección de registros.
4. Mostramos cómo llamar a la función y mostrar los resultados en un bloque PL/SQL.
5. REALMENTE TODO DEBE IR DENTOR DE UN BLOQUEPL/SQL SINO NO FUNCIONA
6. AL FINAL UN EJEMPLO REAL DE COMO SERIAAAA

Este enfoque es muy básico y utiliza conceptos estándar de PL/SQL como tipos de objetos y colecciones. Si tienes alguna pregunta o quieres personalizarlo más, no dudes en preguntar.

**PASOS SI USO EL ROWTYPE COMO ESTRUCTURA**

Cómo usar %ROWTYPE correctamente:

1. Definir un tipo de registro con %ROWTYPE: Para crear un tipo de registro que coincida con la estructura de una tabla, puedes usar el tipo %ROWTYPE.
2. Colección de registros: Luego, puedes crear una colección (un tipo de tabla) que contenga esos registros.

Ejemplo de código usando %ROWTYPE:

**Paso 1: Crear la colección con %ROWTYPE**

/\* Crear el tipo de colección de registros usando %ROWTYPE \*/

CREATE OR REPLACE TYPE TABLA\_EMPLEADOS IS TABLE OF EMP%ROWTYPE;

/

Aquí estamos creando un tipo de colección llamado TABLA\_EMPLEADOS, que es una tabla (colección) de registros. Cada registro tiene el mismo tipo que una fila de la tabla EMP, lo que significa que automáticamente tiene todos los campos de la tabla EMP (sin tener que definir cada uno manualmente).

**Paso 2: Crear la función que devuelve la colección**

/\* Crear la función que devuelve la tabla de empleados \*/

CREATE OR REPLACE FUNCTION FUNCION\_TABLA

RETURN TABLA\_EMPLEADOS

IS

VAUX\_TABLA TABLA\_EMPLEADOS;

BEGIN

-- Usamos BULK COLLECT para llenar la colección con los datos de EMP

SELECT \*

BULK COLLECT INTO VAUX\_TABLA

FROM EMP;

-- Devolvemos la colección

RETURN VAUX\_TABLA;

END FUNCION\_TABLA;

/

Aquí, la función FUNCION\_TABLA selecciona todas las columnas de la tabla EMP y las recoge en la colección VAUX\_TABLA. Dado que VAUX\_TABLA es de tipo TABLA\_EMPLEADOS, puede almacenar registros de la misma estructura que una fila de la tabla EMP.

**Paso 3: Llamar a la función y recorrer los registros**

/\* Llamar a la función y recorrer los registros \*/

DECLARE

VAUX\_TABLA TABLA\_EMPLEADOS;

BEGIN

-- Ejecutamos la función

VAUX\_TABLA := FUNCION\_TABLA();

-- Iteramos sobre la colección de empleados

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

DBMS\_OUTPUT.PUT\_LINE('NUMERO EMPLEADO: ' || VAUX\_TABLA(i).EMPNO || ' NOMBRE: ' || VAUX\_TABLA(i).ENAME);

END LOOP;

END;

/

En este bloque anónimo, llamamos a FUNCION\_TABLA para obtener la colección de registros, y luego usamos un bucle para recorrer la colección y mostrar los valores de los campos de cada registro.

Resumen

* %ROWTYPE te permite crear un tipo de registro que coincide con la estructura de una fila de la tabla, por lo que no necesitas definir cada columna manualmente.
* TABLE OF se usa para crear una colección de registros de este tipo.
* BULK COLLECT carga los datos de la tabla en la colección.

Beneficios de usar %ROWTYPE

* Automatización: No necesitas escribir manualmente cada columna, lo que ahorra tiempo y reduce el riesgo de errores.
* Flexibilidad: Si la estructura de la tabla EMP cambia (por ejemplo, se agregan nuevas columnas), no tendrás que actualizar el código donde uses %ROWTYPE, porque se adaptará automáticamente.

Este enfoque es completamente válido y es una manera muy común de manejar colecciones de registros en PL/SQL.

Si tienes alguna otra pregunta o necesitas más detalles, ¡estoy aquí para ayudarte!

**///////////////////////////////////////////////////ENFOCQUE DE LOS DOS/////////////////////////**

**¡Exacto! Lo has entendido muy bien, y te voy a explicar la diferencia con más detalle para que quede completamente claro:**

**Uso de RECORD vs. ROWTYPE**

1. **Si usas RECORD:**
   * **Primero tienes que definir explícitamente la estructura de las columnas que quieres usar.**
   * **Luego, creas una colección de tipo TABLE OF RECORD si deseas almacenar varias filas (es decir, una colección de registros).**
   * **RECORD se usa cuando necesitas crear una estructura de datos personalizada (por ejemplo, solo algunas columnas de una tabla o campos específicos con tipos específicos).**

**Ejemplo usando RECORD:**

**-- Definir el tipo RECORD**

**DECLARE**

**TYPE emp\_record IS RECORD (**

**empno EMP.EMPNO%TYPE,**

**ename EMP.ENAME%TYPE**

**);**

**-- Definir la colección de tipo RECORD**

**TYPE emp\_table IS TABLE OF emp\_record;**

**-- Variable de tipo colección**

**emp\_collection emp\_table;**

**BEGIN**

**-- Rellenar la colección manualmente o con BULK COLLECT**

**END;**

1. **Si usas ROWTYPE:**
   * **ROWTYPE se utiliza cuando quieres que la estructura del registro coincida exactamente con una fila de una tabla existente. No necesitas definir las columnas manualmente, ya que ROWTYPE toma la estructura de la tabla y crea un tipo de registro automáticamente con todos sus campos.**
   * **Puedes usar TABLE OF ROWTYPE para crear una colección que almacene múltiples registros de una tabla.**

**Ejemplo usando ROWTYPE:**

**-- Crear una colección de registros con %ROWTYPE**

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

**/**

**-- Función que devuelve la colección de registros**

**CREATE OR REPLACE FUNCTION FUNCION\_TABLA**

**RETURN TABLA\_EMPLEADOS**

**IS**

**VAUX\_TABLA TABLA\_EMPLEADOS;**

**BEGIN**

**SELECT \***

**BULK COLLECT INTO VAUX\_TABLA**

**FROM EMP;**

**RETURN VAUX\_TABLA;**

**END FUNCION\_TABLA;**

**/**

**-- Llamar a la función y recorrer los registros**

**DECLARE**

**VAUX\_TABLA TABLA\_EMPLEADOS;**

**BEGIN**

**VAUX\_TABLA := FUNCION\_TABLA();**

**-- Iterar sobre la colección**

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

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

**END LOOP;**

**END;**

**Resumen de las diferencias:**

* **Usando RECORD:**
  + **Paso 1: Definir un tipo RECORD con las columnas que quieres (puedes elegir qué columnas incluir).**
  + **Paso 2: Crear una colección de tipo TABLE OF RECORD si necesitas almacenar varias filas.**
  + **Paso 3: Crear la función que rellena la colección.**
* **Usando ROWTYPE:**
  + **Paso 1: Usar %ROWTYPE para crear un tipo de registro que coincida exactamente con la estructura de una tabla existente.**
  + **Paso 2: Crear una colección de tipo TABLE OF ROWTYPE.**
  + **Paso 3: Crear la función que devuelve esa colección (usando BULK COLLECT para obtener los registros).**

**¿Qué elegir?**

* **Usar RECORD: Cuando necesitas una estructura personalizada, por ejemplo, si solo quieres seleccionar ciertas columnas de una tabla o si deseas un tipo de registro que no se corresponde exactamente con la estructura de una tabla.**
* **Usar ROWTYPE: Cuando deseas que el registro coincida exactamente con la estructura de una tabla (sin tener que definir manualmente cada columna).**

**Conclusión:**

* **Si decides usar ROWTYPE, no necesitas crear un tipo de registro manualmente; es suficiente con usar %ROWTYPE para que automáticamente se cree un tipo de registro que refleje la estructura de la tabla.**
* **Si usas RECORD, necesitarías definir las columnas del registro manualmente y luego crear la colección de ese tipo.**

**Espero que ahora esté todo claro. Si tienes más preguntas o quieres que te dé más ejemplos, ¡avísame!**

**////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**

**EN BLOQUE ANONIMO**

**Ejemplo 1: Usando RECORD dentro de un bloque anónimo**

**Este ejemplo define un RECORD con columnas específicas de la tabla EMP y luego usa BULK COLLECT para cargar datos en una colección de ese tipo.**

**DECLARE**

**-- 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;**

**/**

**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 almacena registros de tipo emp\_record.**
3. **Uso de BULK COLLECT: Se usa para llenar la colección con los resultados de la consulta SELECT EMPNO, ENAME FROM EMP.**
4. **Iteración: Se itera sobre la colección y se imprime el EMPNO y ENAME de cada registro.**

**Ejemplo 2: Usando ROWTYPE dentro de un bloque anónimo**

**Este ejemplo usa %ROWTYPE para crear un registro basado directamente en la estructura de la tabla EMP y luego lo almacena en una colección.**

**DECLARE**

**-- 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;**

**/**

**Explicación:**

1. **Uso de ROWTYPE: Se define emp\_table como una colección de registros de tipo EMP%ROWTYPE. Esto significa que el registro tendrá todas las columnas de la tabla EMP sin necesidad de especificar las columnas manualmente.**
2. **Colección de tipo ROWTYPE: La colección emp\_collection almacenará registros de tipo EMP%ROWTYPE.**
3. **Uso de BULK COLLECT: Al igual que en el ejemplo anterior, se utiliza BULK COLLECT para llenar la colección con todos los registros de la tabla EMP.**
4. **Iteración: Se recorre la colección y se imprimen los valores de EMPNO y ENAME.**

**Comparación:**

* **RECORD te permite crear una estructura personalizada de columnas, mientras que ROWTYPE crea automáticamente un tipo de registro que refleja la estructura completa de una tabla.**
* **En el enfoque con ROWTYPE, no necesitas definir cada columna manualmente, lo que hace que el código sea más limpio y fácil de mantener si trabajas con una tabla entera.**

**Espero que estos ejemplos te hayan aclarado cómo usar ambos enfoques dentro de un bloque anónimo. Si tienes más dudas o preguntas, ¡avísame!**

**//////////////////////**

**ESTO SERIA LO CORRECTO HAY QUE ENGOLBAR DENTRO DE UN DECLARE BEGIN END YA SEA FUNCION O PROCEDIMIENTO SINO NO FUCIONAN LOS RESITGTROS O ROWTYPE**

**DECLARE**

**-- Definir el tipo de colección que se va a usar (TABLE OF EMP%ROWTYPE)**

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

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

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

**Esta es para luego poder llamar a la funcion**

**-- Definir una función que devuelve la colección de registros de EMP**

**FUNCTION FUNCION\_TABLA\_EMPLEADOS RETURN TABLA\_EMPLEADOS IS**

**-- Definir una variable interna para almacenar la colección de registros**

**VAUX\_TABLA TABLA\_EMPLEADOS;**

**BEGIN**

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

**SELECT \***

**BULK COLLECT INTO VAUX\_TABLA**

**FROM EMP;**

**-- Retornar la colección de registros**

**RETURN VAUX\_TABLA;**

**END FUNCION\_TABLA\_EMPLEADOS;**

**BEGIN**

**-- Llamar a la función y almacenar el resultado en la variable VAUX\_TABLA**

**VAUX\_TABLA\_LLAMADA := FUNCION\_TABLA\_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;**

**/**

**///////////////////////////////////////////////////////////////////CON EL 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;**

**-- Definir una función que devuelve la colección de registros**

**FUNCTION FUNCION\_TABLA\_EMPLEADOS RETURN TABLA\_EMPLEADOS IS**

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

**VAUX\_TABLA TABLA\_EMPLEADOS;**

**BEGIN**

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

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

**FROM EMP) LOOP**

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

**VAUX\_TABLA.EXTEND;**

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

**END LOOP;**

**-- Retornar la colección de registros**

**RETURN VAUX\_TABLA;**

**END FUNCION\_TABLA\_EMPLEADOS;**

**BEGIN**

**-- Llamar a la función y almacenar el resultado en la variable VAUX\_TABLA\_LLAMADA**

**VAUX\_TABLA\_LLAMADA := FUNCION\_TABLA\_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;**

**/**