¡Muy buena pregunta, Bryan! 🔥 Vamos a ver **la diferencia entre usar RETURN y un parámetro OUT** en PL/SQL, con ejemplos y explicaciones claras.

**✅ 1. ¿Qué es RETURN?**

Se usa en **funciones** para devolver **un único valor**.  
Una **función** en PL/SQL **siempre devuelve un valor** con RETURN.

**📌 Ejemplo con RETURN:**

CREATE OR REPLACE FUNCTION obtener\_dia(pfecha DATE)

RETURN VARCHAR2

IS

vdia VARCHAR2(10);

BEGIN

vdia := TO\_CHAR(pfecha, 'Day');

RETURN vdia;

END;

/

📌 Se usa así:

DECLARE

resultado VARCHAR2(10);

BEGIN

resultado := obtener\_dia(SYSDATE);

DBMS\_OUTPUT.PUT\_LINE(resultado);

END;

/

👉 **Solo devuelve 1 cosa:** el día como texto.

**✅ 2. ¿Qué es un parámetro OUT?**

Se usa en **procedimientos** (o también funciones, aunque menos común) para **devolver valores por fuera del procedimiento**.  
Puedes tener **varios parámetros OUT** para devolver múltiples valores.

**📌 Ejemplo con OUT:**

CREATE OR REPLACE PROCEDURE obtener\_fecha\_detalle(

pfecha IN DATE,

panio OUT NUMBER,

pmes OUT NUMBER,

pdia OUT NUMBER

)

IS

BEGIN

panio := TO\_NUMBER(TO\_CHAR(pfecha, 'YYYY'));

pmes := TO\_NUMBER(TO\_CHAR(pfecha, 'MM'));

pdia := TO\_NUMBER(TO\_CHAR(pfecha, 'DD'));

END;

/

📌 Se usa así:

DECLARE

anio NUMBER;

mes NUMBER;

dia NUMBER;

BEGIN

obtener\_fecha\_detalle(SYSDATE, anio, mes, dia);

DBMS\_OUTPUT.PUT\_LINE('Año: ' || anio || ', Mes: ' || mes || ', Día: ' || dia);

END;

/

**🎯 Diferencias clave**

| **Característica** | **RETURN en función** | **OUT en procedimiento/función** |
| --- | --- | --- |
| ¿Dónde se usa? | Solo en funciones | En procedimientos o funciones |
| ¿Cuántos valores devuelve? | Solo **uno** | Puede devolver **varios** |
| ¿Se puede usar en SQL? | Sí (ej: SELECT obtener\_valor(...)) | No, solo en bloques PL/SQL |
| ¿Se necesita RETURN? | Sí, obligatorio | No, se usan variables OUT |

**🧠 ¿Cuándo usar cada uno?**

* Usa **RETURN** si necesitas **una función con un solo resultado**, como obtener un año, un total, una media, etc.
* Usa **OUT** si necesitas **varios resultados al mismo tiempo**, como día, mes y año, o múltiples cálculos juntos.

¿Quieres que hagamos una función que use tanto RETURN como un parámetro OUT solo para practicar cómo se puede combinar? 😄

¡Perfecto, Bryan! Vamos a crear una **función que devuelva el año con RETURN** y además devuelva **el mes y el día como parámetros OUT**, todo junto 🔧📅

**✅ Función combinada: RETURN + OUT**

CREATE OR REPLACE FUNCTION analizar\_fecha(

pfecha IN DATE,

pmes OUT NUMBER,

pdia OUT NUMBER

)

RETURN NUMBER

IS

vanio NUMBER;

BEGIN

-- Obtener año

vanio := TO\_NUMBER(TO\_CHAR(pfecha, 'YYYY'));

-- Obtener mes y día por OUT

pmes := TO\_NUMBER(TO\_CHAR(pfecha, 'MM'));

pdia := TO\_NUMBER(TO\_CHAR(pfecha, 'DD'));

-- Devolver el año con RETURN

RETURN vanio;

END;

/

**🧪 Bloque para probarla:**

DECLARE

fecha\_prueba DATE := SYSDATE;

anio NUMBER;

mes NUMBER;

dia NUMBER;

BEGIN

-- Llamamos a la función: devuelve año, y llena mes y día

anio := analizar\_fecha(fecha\_prueba, mes, dia);

DBMS\_OUTPUT.PUT\_LINE('Año: ' || anio);

DBMS\_OUTPUT.PUT\_LINE('Mes: ' || mes);

DBMS\_OUTPUT.PUT\_LINE('Día: ' || dia);

END;

/

**🧠 ¿Qué hicimos aquí?**

* Recibimos una **fecha** (pfecha).
* Devolvemos el **año** con RETURN.
* Devolvemos el **mes y el día** usando los parámetros OUT.

Esto te permite **devolver más de un dato** desde una función, aunque Oracle recomienda en general que si vas a usar muchos OUT, mejor uses un procedimiento. Pero esto es perfecto para **entender cómo se combinan**.

¿Quieres hacerlo ahora usando un RECORD o una tabla? ¿O pasamos a procedimientos con IN OUT también?

¡Exacto, Bryan! 💡 Lo entendiste perfecto.

Cuando usas un **parámetro OUT** en PL/SQL, ese parámetro **no recibe un valor**, sino que **sirve para devolver (almacenar) uno**. Es decir:

**✅ Al llamar a un procedimiento o función con parámetros OUT:**

* **No pasas un valor**.
* **Pasas una variable vacía** que actuará como **contenedor** donde se guardará el resultado.

**📌 Ejemplo rápido:**

Supongamos este procedimiento:

PROCEDURE dame\_nombre(pid IN NUMBER, pnombre OUT VARCHAR2)

Se llama así:

DECLARE

nombre\_empleado VARCHAR2(100);

BEGIN

dame\_nombre(123, nombre\_empleado);

DBMS\_OUTPUT.PUT\_LINE(nombre\_empleado);

END;

🔹 pnombre es OUT, así que **no se le pasa un valor**.  
🔹 Se le pasa la **variable nombre\_empleado** que **queda llena** después del procedimiento.

**⚠️ Entonces, recuerda:**

| **Tipo de parámetro** | **¿Recibe valor?** | **¿Devuelve valor?** | **¿Cómo lo llamas?** |
| --- | --- | --- | --- |
| IN | ✅ Sí | ❌ No | Le pasas un dato |
| OUT | ❌ No | ✅ Sí | Le pasas una **variable vacía** |
| IN OUT | ✅ Sí | ✅ Sí | Le pasas una **variable con valor** y luego se **modifica** |

¿Quieres que prepare un ejemplo con los tres (IN, OUT, IN OUT) para ver cómo funcionan todos juntos en un mismo procedimiento?