

## UA4B. PL/SQL

---

### 4.16.3. Paso de parámetros

---

Ahora vamos a profundizar un poco más en los parámetros que aceptan los subprogramas y cómo se los podemos pasar a la hora de invocarlos.

Las variables pasadas como parámetros a un subprograma son llamadas **parámetros actuales**. Las variables referenciadas en la especificación del subprograma como parámetros, son llamadas **parámetros formales**.

Cuando llamamos a un subprograma, los parámetros actuales podemos escribirlos utilizando notación posicional o notación nombrada, es decir, la asociación entre parámetros actuales y formales podemos hacerla por posición o por nombre.

En la notación posicional, el primer parámetro actual se asocia con el primer parámetro formal, el segundo con el segundo, y así para el resto. En la notación nombrada usamos el operador => para asociar el parámetro actual al parámetro formal. También podemos usar notación mixta.

Los parámetros pueden ser de **entrada** al subprograma, de **salida**, o de **entrada/salida**. Por defecto si a un parámetro no le especificamos el modo, éste será de entrada. Si el parámetro es de salida o de entrada/salida, el parámetro actual debe ser una variable.

Un parámetro de entrada permite que pasemos valores al subprograma y no puede ser modificado en el cuerpo del subprograma. El parámetro actual pasado a un subprograma como parámetro formal de entrada puede ser una constante o una variable.

Un parámetro de salida permite devolver valores y dentro del subprograma actúa como variable no inicializada. El parámetro formal debe ser siempre una variable.

Un parámetro de entrada-salida se utiliza para pasar valores al subprograma y/o para recibirlos, por lo que un parámetro formal que actúe como parámetro actual de entrada-salida siempre deberá ser una variable.

Los parámetros de entrada los podemos inicializar a un valor por defecto. Si un subprograma tiene un parámetro inicializado con un valor por defecto, podemos invocarlo prescindiendo del parámetro y aceptando el valor por defecto o pasando el parámetro y sobrescribiendo el valor por defecto. Si queremos prescindir de un parámetro colocado entre medias de otros, deberemos usar notación nombrada o si los parámetros restantes también tienen valor por defecto, omitirlos todos.

## Ejemplos

---

Veamos algunos ejemplos del paso de parámetros a subprogramas:

### Notación mixta

```
DECLARE
PROCEDURE prueba( formal1 NUMBER, formal2 VARCHAR2) IS
BEGIN
...
END;
actual1 NUMBER;
actual2 VARCHAR2;
BEGIN
...
prueba(actual1, actual2); --posicional
prueba(formal2=>actual2, formal1=>actual1); --nombrada
prueba(actual1, formal2=>actual2); --mixta
END;
```

## Parámetros de entrada

```
FUNCTION categoria( id_agente IN NUMBER )  
RETURN NUMBER IS  
cat NUMBER;  
BEGIN  
...  
SELECT categoria INTO cat FROM agentes  
WHERE identificador = id_agente;  
RETURN cat;  
EXCEPTION  
WHEN NO_DATA_FOUND THEN  
id_agente := -1; --ilegal, parámetro de entrada  
END;
```

## Parámetros de salida

```
PROCEDURE nombre( id_agente NUMBER, nombre OUT VARCHAR2) IS  
BEGIN  
IF (nombre = 'LUIS') THEN --error de sintaxis  
END IF;  
...  
END;
```

Obra publicada con **Licencia Creative Commons Reconocimiento Compartir igual 4.0**  
<<http://creativecommons.org/licenses/by-sa/4.0/>>

