

UA4B. PL/SQL

4.13.1. Estructuras de control

En la vida constantemente tenemos que tomar decisiones que hacen que llevemos a cabo unas acciones u otras dependiendo de unas circunstancias o repetir una serie de acciones un número dado de veces o hasta que se cumpla una condición. En PL/SQL también podemos imitar estas situaciones por medio de las estructuras de control que son sentencias que nos permiten manejar el **flujo de control** de nuestro programa, y éstas son dos: condicionales e iterativas.

Las **estructuras de control condicional** nos permiten llevar a cabo una acción u otra dependiendo de una condición. Vemos sus diferentes variantes:

IF-THEN

Forma más simple de las sentencias de control condicional. Si la evaluación de la condición es **TRUE**, entonces se ejecuta la secuencia de sentencias encerradas entre el **THEN** y el final de la sentencia.

Sentencia IF-THEN.

Sintaxis.	Ejemplo.
<pre>IF condicion THEN secuencia_de_sentencias; END IF;</pre>	<pre>IF (b<>0) THEN c:=a/b; END IF;</pre>

IF-THEN-ELSE

Con esta forma de la sentencia ejecutaremos la primera secuencia de sentencias si la condición evalúa a **TRUE** y en caso contrario ejecutaremos la segunda secuencia de sentencias.

Sentencia IF-THEN-ELSE.

Sintaxis.	Ejemplo.
<pre>IF condicion THEN Secuencia_de_sentencias1; ELSE Secuencia_de_sentencias2; END IF;</pre>	<pre>IF (b<>0) THEN c:=a/b; END IF;</pre>

IF-THEN-ELSIF

Con esta última forma de la sentencia condicional podemos hacer una selección múltiple. Si la evaluación de la condición 1 da **TRUE**, ejecutamos la secuencia de sentencias 1, sino evaluamos la condición 2. Si esta evalúa a **TRUE** ejecutamos la secuencia de sentencias 2 y así para todos los **ELSIF** que haya. El último **ELSE** es opcional y es por si no se cumple ninguna de las condiciones anteriores.

Sentencia IF-THEN-ELSIF.

Sintaxis.	Ejemplo.
<pre>IF condicion1 THEN Secuencia_de_sentencias1; ELSIF condicion2 THEN</pre>	<pre>IF (operacion = 'SUMA') THEN resultado := arg1 + arg ELSIF (operacion = 'RESTA')</pre>

Sintaxis.	Ejemplo.
<pre> Secuencia_de_sentencias2; ... [ELSE Secuencia_de_sentencias;] END IF;</pre>	<pre> resultado := arg1 - arg ELSIF (operacion = 'PRODUCTO resultado := arg1 * arg ELSIF (arg2 <> 0) AND (opera resultado := arg1 / arg ELSE RAISE operacion_no_perm END IF;</pre>

CASE

Se trata de una sentencia condicional múltiple. Tiene la misma funcionalidad que una sentencia IF-THEN-ELSE

Sentencia CASE.

Sintaxis.	Ejemplo.
<pre> CASE [expresion] WHEN condicion1 THEN resultado1; WHEN condicion2 THEN resultado2; ... WHEN condicionN THEN resultadoN; ELSE resultado; END CASE;</pre>	<pre> CASE n WHEN 1 THEN dbms_output.p WHEN 2 THEN dbms_output.p WHEN 2+2 THEN dbms_output ELSE dbms_output.put_line END CASE;</pre>

Las estructuras de control iterativo permiten ejecutar una secuencia de sentencias un determinado número de veces.

LOOP

La forma más simple es el bucle infinito, cuya sintaxis es:

```
LOOP
  secuencia_de_sentencias;
END LOOP;
```

EXIT

Con esta sentencia forzamos a un bucle a terminar y pasa el control a la siguiente sentencia después del bucle. Un EXIT no fuerza la salida de un bloque PL/SQL, sólo la salida del bucle.

```
LOOP
  ...
  IF encontrado = TRUE THEN
    EXIT;
  END IF;
END LOOP;
```

EXIT WHEN condicion

Fuerza a salir del bucle cuando se cumple una determinada condición.

```
LOOP
  ...
  EXIT WHEN encontrado;
END LOOP;
```

WHILE LOOP

Este tipo de bucle ejecuta la secuencia de sentencias mientras la condición sea cierta.

```
WHILE condicion LOOP
    Secuencia_de_sentencias;
END LOOP;
```

```
WHILE (not encontrado) LOOP
    ...
END LOOP;
```

FOR LOOP

Este bucle itera mientras el contador se encuentre en el rango definido.

```
FOR contador IN [REVERSE] limite_inferior..limite_superior LOOP
    ...
END LOOP; /* Otro ejemplo */
FOR i IN 1..3 LOOP --i=1, i=2, i=3
    ...
END LOOP;
SELECT count(*) INTO num_agentes FROM agentes;
FOR i IN 1..num_agentes LOOP
    ...
END LOOP;

FOR i IN REVERSE 1..3 LOOP --i=3, i=2, i=1
    ...
END LOOP;
```