

Bloques PL/SQL

Un programa de PL/SQL está compuesto por bloques. Un programa está compuesto como mínimo de un bloque.

Los bloques de PL/SQL pueden ser de los siguientes tipos:

- Bloques anónimos
- Subprogramas

Estructura de un Bloque

Los bloques PL/SQL presentan una estructura específica compuesta de tres partes bien diferenciadas:

- La sección declarativa en donde se declaran todas las constantes y variables que se van a utilizar en la ejecución del bloque.
- La sección de ejecución que incluye las instrucciones a ejecutar en el bloque PL/SQL.
- La sección de excepciones en donde se definen los manejadores de errores que soportará el bloque PL/SQL.

Cada una de las partes anteriores se delimita por una palabra reservada, de modo que un bloque PL/SQL se puede representar como sigue:

```
declare
/*Parte declarativa*/
begin
/*Parte de ejecucion*/
exception
/*Parte de excepciones*/
end;
```

De las anteriores partes, únicamente la sección de ejecución es obligatoria, que quedaría delimitada entre las cláusulas **BEGIN** y **END**. Veamos un ejemplo de bloque PL/SQL muy genérico. Se trata de un bloque anónimos, es decir no lo identifica ningún nombre. Los bloques anónimos identifican su parte declarativa con la palabra reservada **DECLARE**.

```
DECLARE
/*Parte declarativa*/
    nombre_variable DATE;
BEGIN
/*Parte de ejecucion
* Este código asigna el valor de la columna "nombre_columna"
* a la variable identificada por "nombre_variable"
*/
    SELECT SYSDATE INTO nombre_variable FROM DUAL;

EXCEPTION
```

```

/*Parte de excepciones*/
WHEN OTHERS THEN
    dbms_output.put_line('Se ha producido un error');
END;

```

A continuación vamos a ver cada una de estas secciones

Sección de Declaración de Variables

En esta parte se declaran las variables que va a necesitar nuestro programa. Una variable se declara asignándole un nombre o "identificador" seguido del tipo de valor que puede contener. También se declaran cursores, de gran utilidad para la consulta de datos, y excepciones definidas por el usuario. También podemos especificar si se trata de una constante, si puede contener valor nulo y asignar un valor inicial.

La sintaxis generica para la declaracion de constantes y variables es:

```

nombre [constant] tipo [not null][:=valor];

```

donde:

- **tipo_dato:** es el tipo de dato que va a poder almacenar la variable, este puede ser cualquiera de los tipos soportados por ORACLE, es decir NUMBER, DATE, CHAR, VARCHAR, VARCHAR2, BOOLEAN ... Además para algunos tipos de datos (NUMBER y VARCHAR) podemos especificar la longitud.
- La cláusula **CONSTANT** indica la definición de una constante cuyo valor no puede ser modificado. Se debe incluir la inicialización de la constante en su declaración.
- La cláusula **NOT NULL** impide que a una variable se le asigne el valor nulo, y por tanto debe inicializarse a un valor diferente de NULL.
- Las variables que no son inicializadas toman el valor inicial NULL.
- La inicialización puede incluir cualquier expresión legal de PL/SQL, que lógicamente debe corresponder con el tipo del identificador definido.
- Los tipos escalares incluyen los definidos en SQL más los tipos VARCHAR y BOOLEAN. Este último puede tomar los valores TRUE, FALSE y NULL, y se suele utilizar para almacenar el resultado de alguna operación lógica. VARCHAR es un sinónimo de CHAR.
- También es posible definir el tipo de una variable o constante, dependiendo del tipo de otro identificador, mediante la utilización de las cláusulas %TYPE y %ROWTYPE. Mediante la primera opción se define una variable o constante escalar, y con la segunda se define una variable fila, donde identificador puede ser otra variable fila o una tabla. Habitualmente se utiliza %TYPE para definir la variable del mismo tipo que tenga definido un campo en una tabla de la base de datos, mientras que %ROWTYPE se utiliza para declarar variables utilizando cursores.