

- Los **tipos de datos escalares** estudiados son: *carácter* (CHAR, NCHAR, VARCHAR2, NVARCHAR2...), *numérico* (NUMBER, BINARY\_INTEGER, PLS\_INTEGER), *booleano*, *Fecha/hora* (DATE, TIMESTAMP...) y otros ( ROWID, UROWID,...).

- Los **identificadores** en PL/SQL pueden tener entre 1 y 30 caracteres de longitud; el primer carácter debe ser una letra y los restantes deben ser caracteres alfanuméricos o signos admitidos (letras, dígitos, los signos de dólar, almohadilla y subguión); no pueden incluir signos de puntuación, espacios, etcétera.

- El **formato para declarar una variable** es:  
`<nombre_de_variable> <tipo> [NOT NULL]  
 [{:= | DEFAULT} <valor>]`

- Las variables se crean al comienzo del bloque y dejan de existir una vez finalizada la ejecución del bloque en el que han sido declaradas. El ámbito de una variable incluye el bloque en el que se declara y los bloques «hijos» de este.

- Al declarar una constante deberemos incluir CONSTANT y asignar un valor.

- En PL/SQL, podemos insertar comentarios: de línea con -- o de varias líneas con /\* ... \*/.

- Las **estructuras de control** son:

- *Alternativa simple*: IF ... THEN ...; END IF;
- *Alternativa doble*: IF ... THEN ...; ELSE ...; END IF;

- *Alternativa múltiple*: IF ... THEN ...; ELSIF ... THEN ...; ELSIF ... THEN ...; ... ELSE ...; END IF;

- *Alternativa multiple*: CASE ... WHEN ... THEN ...; WHEN ... THEN ...; ... ELSE ...; END CASE;

- *Iterar*: LOOP ...; EXIT WHEN ...; END LOOP;

- *Mientras*: WHILE ... LOOP ...; END LOOP;
- *Para*: FOR I in 1.. n LOOP ...; END LOOP;

- Para crear un *procedimiento* o *función* desde SQL\*Plus usaremos CREATE OR REPLACE ...

```
PROCEDURE <nombreprocedimiento>
  [( <lista de parámetros> )]
IS  ... < BLOQUE PL/SQL > ;
```

```
FUNCTION <nombrefunción>
  [( <lista de parámetros> )]
RETURN <tipo d valor devuelto >
IS  ... < BLOQUE PL/SQL que incluye
RETURN <expresión>>
```

- La instrucción para invocar un procedimiento es un comando en sí misma. En el caso de las funciones es una expresión que debe hacer algo con el valor devuelto.

- Al indicar los parámetros debemos especificar el tipo pero no el tamaño. En el caso de que el subprograma no tenga parámetros no se pondrán los paréntesis.

- Los parámetros en modo IN, dentro del subprograma, actúan como una constante, es decir, no se les puede asignar ningún otro valor.

1 Escribe un procedimiento que reciba dos números y visualice su suma.

2 Codifica un procedimiento que reciba una cadena y la visualice al revés.

3 Reescribe el código de los dos ejercicios anteriores para convertirlos en funciones que retornen los valores que mostraban los procedimientos.

4 Escribe una función que reciba una fecha y devuelva el año, en número, correspondiente a esa fecha.

5 Escribe un bloque PL/SQL que haga uso de la función anterior.

6 Desarrolla una función que devuelva el número de años completos que hay entre dos fechas que se pasan como parámetros.

7 Escribe una función que, haciendo uso de la función anterior, devuelva los trienios que hay entre dos fechas (un trienio son tres años).

8 Codifica un procedimiento que reciba una lista de hasta cinco números y visualice su suma.

9 Escribe una función que devuelva solamente caracteres alfabéticos sustituyendo cualquier otro carácter por blancos a partir de una cadena que se pasará en la llamada.

10 Codifica un procedimiento que permita borrar un empleado cuyo número se pasará en la llamada.

11 Escribe un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la nueva localidad.

12 Visualiza todos los procedimientos y funciones del usuario almacenados en la base de datos y su situación (*valid* o *invalid*).