

nuevos tipos de datos sino que, por el contrario, permiten establecer restricciones sobre tipos de datos existentes.

El objetivo de los subtipos es simplificar la escritura y facilitar la comprensión del programa, empleando nombres de tipos de datos conocidos por el programador o utilizados en la empresa. Por ejemplo, el tipo de datos CHARACTER es un subtipo de CHAR que permite que el lenguaje PL/SQL respete la norma ANSI.

Después de definir los subtipos en la sección DECLARE del bloque PL/SQL, se pueden definir en él variables del mismo modo que se definen variables pertenecientes a tipos de la base de datos.

Sintaxis

```
SUBTYPE nombre_subtipo IS tipo;
```

Ejemplo

Definición y uso de un subtipo: declaración del subtipo fechanacimiento y uso en un bloque PL/SQL.

```
SQL> declare
  2  subtype fechanacimiento is date;
  3  nac fechanacimiento;
  4  begin
  5      -- asignar la fecha del día a la variable nac
  6      select sysdate
  7      into nac
  8      from dual;
  9  end;
 10  /
```

Procedimiento PL/SQL terminado correctamente.

```
SQL>
```

4. Colecciones y registros

Las colecciones y los registros tienen el objetivo de facilitar la programación en PL/SQL. Estos tipos de datos sólo existen en PL/SQL y no tienen equivalente en la base de datos Oracle. No es posible almacenar directamente un registro en la base de datos, pero, sin embargo, sí se puede almacenar cada elemento que compone el registro en la base de datos.

a. Colecciones

Una colección es un conjunto ordenado de elementos del mismo tipo. Una colección puede compararse con las matrices que pueden definirse en ciertos lenguajes de programación, con la diferencia de que las colecciones sólo pueden tener una dimensión y los elementos de la misma se indexan mediante un valor de tipo numérico.


Entre las colecciones se distinguen dos tipos diferentes: las tablas (TABLE) de tipo *index by table* (índice por tabla) o *nested table* (tabla anidada), que amplían las funcionalidades ofrecidas por las tablas de tipo índice, y las tablas de tipo VARRAY.

Después de la declaración, las colecciones tienen que ser inicializadas. Para ello, existe una función específica, denominada constructor, que usa el mismo nombre que la colección y acepta como parámetros los valores que debe contener inicialmente la colección. Precaución: Oracle no asigna implícitamente un nombre a este constructor.

Colecciones de tipo tabla anidada e índice por tabla

Estas colecciones tienen tamaño dinámico y no existen forzosamente valores para todas las posiciones.

10		75	38		74	37	44			01	64
x(1)		x(3)	x(4)		x(7)	x(8)	x(9)			x(13)	x(14)



Las colecciones

Como muestra el esquema anterior, las colecciones son mucho más flexibles que las tablas, ya que se pueden eliminar determinados elementos de la colección.

Las colecciones de tipo tabla anidada presentan la particularidad de que los datos que contienen pueden almacenarse directamente en una columna de tabla, de ahí su nombre. Además, las colecciones no inicializadas toman el valor nulo (NULL).

Las colecciones de tipo índice por tabla representan el método más rápido para transferir datos entre un bloque PL/SQL y la base de datos.

Sintaxis

Declaración de una colección de tipo tabla anidada

```
TYPE nombre_tipo IS TABLE OF tipo_elemento [NOT NULL]
```

Declaración de una colección de tipo *índice por tabla*

```
TYPE nombre_tipo IS TABLE OF tipo_elemento [NOT NULL]  
INDEX BY BINARY_INTEGER
```

Ejemplo

Declaración y uso de colecciones: en el siguiente ejemplo se definen dos colecciones y luego se emplean para validar su definición (el uso de colecciones se describe en detalle más adelante en el capítulo).

```
SQL> declare  
2   -- colección de tipo tabla anidada  
3   type cuadrícula is table of number;  
4   -- colección de tipo tabla organizada por índice (index by)  
5   type citas is table of date index by binary_integer;  
6   lotería cuadrícula;  
7   agenda citas;  
8   begin  
9   -- números a los que jugar  
10  lotería:=cuadrícula(44,37,25,17);  
11  -- anotar una cita  
12  agenda(2):=to_date('27/05/2003 11:30','DD/MM/YYYY HH24:MI');  
13  end;  
14  /
```

Procedimiento PL/SQL terminado correctamente.

SQL>

Colecciones de tipo VARRAY

Una colección de tipo VARRAY tiene una dimensión máxima que debe especificarse al declarar la colección. Estas colecciones tienen longitud fija y, por tanto, la eliminación de elementos no permite ahorrar espacio en memoria. Los elementos se numeran a partir del valor 1.

Sintaxis

```
TYPE nombre_tipo IS VARRAY (tamaño_max) OF tipo_elemento  
[NOT NULL]
```

nombre_tipo

Representa el nombre de la colección.

tamaño_max

Representa el número máximo de elementos presentes en la colección.

tipo_elemento

Representa el tipo de datos de los elementos que constituyen la colección.

Ejemplo

Declaración y uso de colecciones. En el siguiente ejemplo se define una colección denominada Calendario.

```
SQL> declare  
2   type calendario is varray(366) of date;  
3   -- declaración e inicialización de la colección  
4   año calendario:=calendario(to_date('01/01/2003','DD/MM/YYYY'));  
5   begin  
6   -- modificación del valor  
7   año(1):=to_date('02/02/2003','DD/MM/YYYY');  
8   end;  
9   /
```

Procedimiento PL/SQL terminado correctamente.

SQL>

b. Registros

Un registro es un conjunto de valores relacionados entre sí y agrupados bajo un mismo nombre. Cada valor se almacena en un campo. Por ejemplo, el registro que permite manipular un cliente estará formado por los campos número, nombre, dirección... Por tanto, los registros proporcionan un método sencillo de manipular datos que están relacionados entre sí.

Para poder crear un registro, es necesario definir previamente el tipo de registro en la sección DECLARE del bloque PL/SQL.