ÍNDICE

FUNDAMENTOS DEL LENGUAJE PL/SQL 1

INTRODUCCIÓN

PL/SQL es un sofisticado lenguaje de programación que se utiliza para acceder a bases de datos Oracle desde distintos entornos. PL/SQL está integrado con el servidor de base de datos, de modo que el código puede ser procesado de forma rápida y eficiente. También se encuentra disponible en varias de las herramientas de cliente que posee Oracle, entre ellas SQL*PLUS, Developer Suite, JDeveloper, etc.

Si nos preguntamos por qué utilizar PL/SQL, la conclusión la encontramos en el propio SQL. Tenemos que recordar que Oracle es una base de datos relacional, que utiliza como lenguaje de datos el propio SQL. SQL es un lenguaje flexible y eficiente, con características muy potentes para la manipulación y examen de los datos relacionales, pero que presenta deficiencias a la hora de realizar programaciones procedimentales.

SQL es un lenguaje de cuarta generación (4GL), que como el resto de lenguajes de esta generación, presenta como característica el hecho de que describen lo que debe hacerse, pero no la manera de llevarlo a cabo. Por ejemplo si analizamos la siguiente instrucción:

```
DELETE FROM estudiantes WHERE nombre like 'Pep%'
```

Esta instrucción determina que queremos borrar de la tabla estudiantes todos aquellos cuyo nombre comience por "Pep", pero no dice cómo va a realizar el gestor de base de datos el proceso para conseguir eliminar dichos registros. Parece presumible que recorrerá los datos de dicha tabla en un cierto orden para determinar qué elementos debe borrar y luego los eliminará; no obstante, es algo que no nos interesa para la instrucción.

En contraposición a los lenguajes 4GL nos encontramos con los lenguajes de tercera generación (3GL), como C y Visual Basic. Son lenguajes más procedimentales donde se implementan algoritmos para resolver unos problemas. Estas estructuras procedimentales y de ejecución paso a paso no se pueden implementar en SQL, así que Oracle necesitaba de un lenguaje que pudiese resolver este tipo de problemas y que estuviera más enfocado no solo al manejo de datos sino a la resolución de problemáticas de todo tipo, así que creó el lenguaje PL/SQL (Lenguaje Procedimental / SQL). Este lenguaje no es solo un lenguaje de tipo 3GL, sino que permite utilizar la flexibilidad de SQL como lenguaje de 4GL.

Esta característica que lo define, es posible dado que es un lenguaje particular del sistema gestor de bases de datos Oracle y no un lenguaje estándar.

Por tanto el lenguaje PL/SQL potencia el lenguaje SQL agregando estructuras y objetos del siguiente tipo:

- El bloque
- Manejo de errores y excepciones
- Creación de procedimientos y funciones
- Definición de variables y tipos
- Estructuras de bucle
- Cursores
- Objetos

El bloque

Es la unidad básica de todo programa en PL/SQL. Todo programa al menos debe poseer un bloque.

Todo bloque consta de una sección declarativa optativa, una sección de ejecución obligatoria y una sección de control de errores optativa.

La sintaxis es la siguiente:

```
[DECLARE]
        <sección declarativa>
BEGIN
        <sección de ejecución>
[EXCEPTION]
        <sección de control de errores>
END;
```

SECCIÓN DECLARATIVA

En esta sección se definen las variables, constantes y cursores que se van a utilizar dentro de la sección de ejecución.

SECCIÓN DE EJECUCIÓN

La sección de ejecución presenta las siguientes particularidades:

- Toda instrucción `SELECT` (no dinámica) que aparezca en esta sección, deberá llevar incorporado el parámetro `INTO`, como se muestra en el siguiente ejemplo:

  `SELECT` columnas `INTO` variables `FROM` tablas `WHERE` criterios.

- En esta sección solo se admiten instrucciones de SQL del tipo DML (select, insert, update y delete) o instrucciones SQL dinámicas, el resto no están permitidas implementarlas directamente (por ejemplo: alter, create, drop, etc.), salvo que se indique dentro de la instrucción `EXECUTE INMEDIATE`.

SECCIÓN DE CONTROL DE ERRORES

En esta sección se definen los controles programados para detectar los errores de ejecución del bloque y la solución adoptada para los mismos.

Tipos de bloque

Se diferencian 3 tipos de bloques:

- Los *bloques anónimos* se construyen de manera dinámica y se ejecutan una sola vez.
- Los *bloques nominados* se construyen identificándolos con un nombre. Al igual que los anteriores, se construyen de forma dinámica y se ejecutan una sola vez.
- Los *subprogramas* son bloques nominados que se almacenan en la base de datos. Nos podemos encontrar con procedimientos, paquetes y funciones de este tipo. Siempre se ejecutan bajo demanda.
- Los *disparadores* son también bloques nominados que se almacenan en la base de datos, pero que no se pueden ejecutar bajo petición de un programa. Se ejecutan cuando tiene efecto el suceso para el que se han programado contra una cierta tabla del sistema.

EJEMPLO DE BLOQUE ANÓNIMO

```
DECLARE
   Var1  NUMBER;
BEGIN
   Var1 := 1;
END;
```

EJEMPLO DE BLOQUE NOMINADO

```
<<Nombre_Bloque>>
DECLARE
   Var1  NUMBER;
BEGIN
   Var1 := 1;
END;
```

EJEMPLO DE BLOQUE SUBPROGRAMA

```
CREATE OR REPLACE PROCEDURE MI_PROGRAMA IS
   Var1  NUMBER;
BEGIN
   Var1 := 1;
END;
```

EJEMPLO DE DISPARADOR

```
CREATE OR REPLACE TRIGGER MI_DISPARADOR IS
   BEFORE INSERT OR UPDATE OF numero ON tabla_temporal
   FOR EACH ROW
BEGIN
   IF :new.numero < 0 THEN
        RAISE_APPLICATION_ERROR(-20100,'¡¡¡¡Error!!!');
END;
```

Manejo de errores y excepciones

El lenguaje PL/SQL, como muchos otros procedimentales, permite un control sobre los errores que se produzcan en la ejecución del código. Esta gestión de errores presenta como ventaja la claridad para su manipulación, dado que utiliza una sección independiente a la del código ejecutable.

Creación de procedimientos y funciones

El lenguaje PL/SQL permite la creación de procedimientos almacenados y de funciones que nos devuelvan un valor como resultado de su ejecución.

Definición de variables y tipos

El lenguaje PL/SQL también permite la definición de variables para utilizar en nuestros programas y crear tipos de usuarios a partir de otros predefinidos.

Estructuras de bucle

Para desarrollar nuestros programas y poder realizar operaciones de bifurcación, el lenguaje PL/SQL posee estructuras de control.

Cursores

Este tipo de estructura, que se define en la sección declarativa de un bloque, nos permite recuperar en memoria un conjunto de filas de una tabla que se recorren una a una para un tratamiento posterior.

Objetos

Oracle dada la tendencia actual de los lenguajes de programación que hay en el mercado, incorpora también la figura del objeto, de forma que PL/SQL se puede convertir también en un lenguaje orientado a objetos.

Salida por pantalla de los resultados de una ejecución

PL/SQL a través de las herramientas propias de Oracle: SQL*Plus y SQL*Plus Worksheet únicamente visualiza el resultado satisfactorio o no de la ejecución de las instrucciones. Para poder realizar una visualización en pantalla de la ejecución y resultados internos del código PL/SQL hay que utilizar la invocación de un paquete incluido en Oracle PL/SQL denominado DBMS_OUTPUT, el cual permite dirigir a pantalla resultados mediante el uso de la función PUT_LINE.

No obstante para que se haga completamente efectiva dicha salida a pantalla, antes hay que activar el comando de ejecución en pantalla del paquete DBMS_OUTPUT, siempre que se inicie una nueva sesión con la herramienta correspondiente. Este comando es el siguiente:

```
SET SERVEROUTPUT ON;
```

EJEMPLO DE UN BLOQUE CON SALIDA A PANTALLA DE LOS RESULTADOS

```
SET SERVEROUTPUT ON;
DECLARE
   Var1  VARCHAR2(50);
BEGIN
   Var1 := 'Antolin';
   DBMS_OUTPUT.PUT_LINE('El nombre del autor es:'||
                        Var1);
END;
/
```

UNIDADES LÉXICAS

Es el conjunto de caracteres y gramática a utilizar en la programación de PL/SQL. Contamos con los siguientes elementos:

- Identificadores
- Palabras reservadas
- Delimitadores
- Literales
- Comentarios

Identificadores

Dan nombre a los distintos objetos de nuestro programa. Por ejemplo, los identificadores de variable, cursores, tipos, etc.

Un identificador tiene que comenzar obligatoriamente con una letra seguida por una secuencia de caracteres, entre los que se pueden incluir:

- Letras
- Números
- El símbolo $
- El carácter de subrayado _
- El símbolo #

La longitud máxima de un identificador es de 30 caracteres.

También hay que tener en cuenta que el lenguaje PL/SQL no es un lenguaje sensible en cuanto a los caracteres, de manera que no distingue mayúsculas o minúsculas, salvo que el nombre vaya encerrado entre comillas dobles (").

Cada objeto tiene un espacio de nombres.

Los espacios de nombres de objeto aglutinan una serie de nombres de objeto, en algunos casos de forma independiente, y en otros común para distintos nombres de objetos diferentes.

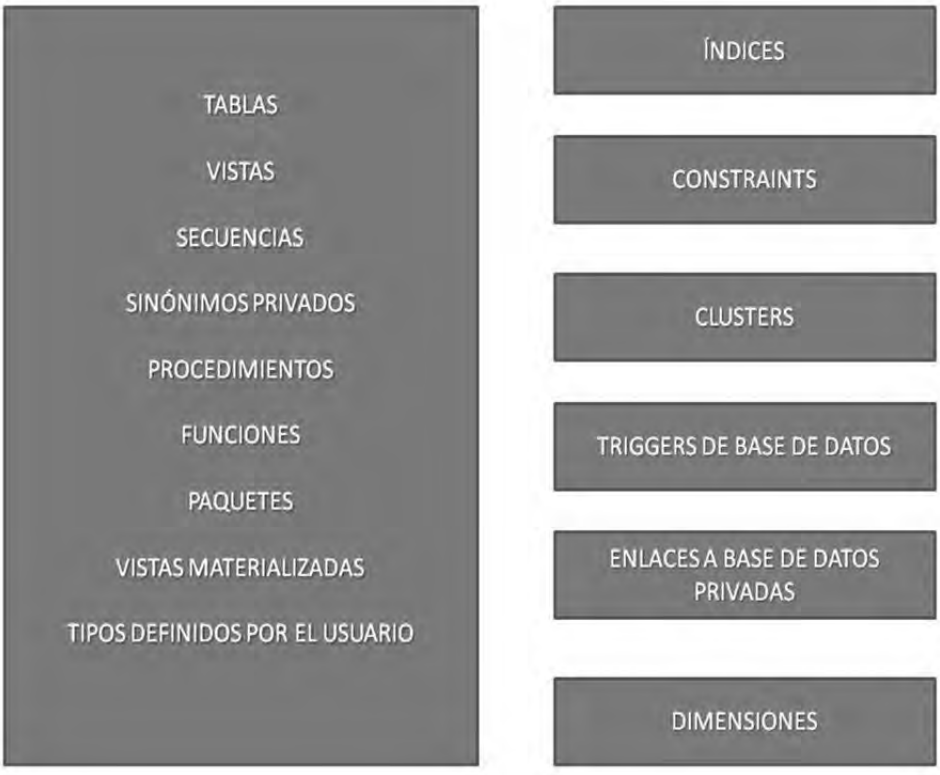

Fig. 1-1 Namespaces (Esquemas de nombre) de objetos de un esquema.

Dentro del mismo espacio de nombres no se puede repetir un nombre de objeto aunque sea de distinto tipo.

Por ejemplo, una tabla y una vista no se pueden denominar con el mismo nombre, porque comparten el mismo esquema de nombres. En cambio una restricción y un trigger sí se pueden llamar igual, porque cada uno de estos objetos del esquema de base de datos poseen esquemas independientes.

Fig. 1-2 Namespaces de objetos que no pertenecen a un esquema.

Ejemplos de nombres de esquema válidos son los siguientes:

```
EMP
"Emp"
SCOTT.FECHAALTA
"INCLUSO ESTO & VALE!"
UN_NOMBRE_LARGO_Y_VALIDO
```

Palabras reservadas

Son todas aquellas que define Oracle como restringidas dentro del lenguaje PL/SQL y cuyo nombre no podrá llevar ningún otro objeto de la base de datos.