

UA4B. PL/SQL

4.16.2. Almacenar subprogramas

Para almacenar un subprograma en la base de datos utilizaremos la misma sintaxis que para declararlo, anteponiendo **CREATE [OR REPLACE]** a **PROCEDURE** o **FUNCTION**, y finalizando el subprograma con una línea que simplemente contendrá el carácter **'/'** para indicarle a Oracle que termina ahí.

Si especificamos **OR REPLACE** y el subprograma ya existía, éste será reemplazado. Si no lo especificamos y el subprograma ya existe, Oracle nos devolverá un error indicando que el nombre ya está siendo utilizado por otro objeto de la base de datos.

```
CREATE OR REPLACE FUNCTION hijos_familia(id_familia NUMBER)
RETURN NUMBER IS
hijos NUMBER;
BEGIN
SELECT COUNT(*) INTO hijos FROM agentes
WHERE familia = id_familia;
RETURN hijos;
END;
/
```

Cuando los subprogramas son almacenados en la base de datos, para ellos no podemos utilizar las declaraciones hacia delante, por lo que cualquier subprograma almacenado en la base de datos deberá conocer todos los subprogramas que utilice.

Para invocar un subprograma usaremos la sintaxis:

```
nombre_procedimiento [(parametro [,parametro] ...)];  
variable := nombre_funcion [(parametro [, parametro] ...)];  
  
BEGIN  
...  
hijos := hijos_familia(10);  
...  
END;
```

Si el subprograma está almacenado en la base de datos y queremos invocarlo desde SQL*Plus usaremos la sintaxis:

```
EXECUTE nombre_procedimiento [(parametros)];  
EXECUTE :variable_sql := nombre_funcion [(parametros)];
```

Cuando almacenamos un subprograma en la base de datos éste es compilado antes. Si hay algún error se nos informará de los mismos y deberemos corregirlos por medio de la cláusula **OR REPLACE**, antes de que el subprograma pueda ser utilizado.

Hay varias vistas del diccionario de datos que nos ayudan a llevar un control de los subprogramas, tanto para ver su código, como los errores de compilación. También hay algunos comandos de SQL*Plus que nos ayudan a hacer lo mismo pero de forma algo menos engorrosa.

Vistas y comandos asociados a los subprogramas.

Información almacenada.	Vista del diccionario.	Comando.
Código fuente.	USER_SOURCE	DESCRIBE
Errores de compilación.	USER_ERRORS	SHOW ERRORS
Ocupación de memoria.	USER_OBJECT_SIZE	

También existe la vista **USER_OBJECTS** de la cual podemos obtener los nombres de todos los subprogramas almacenados.

Obra publicada con **Licencia Creative Commons Reconocimiento Compartir igual 4.0**
<<http://creativecommons.org/licenses/by-sa/4.0/>>