



3. Procediments i funcions

RA5.Desenvolupa procediments emmagatzemats avaluant i utilitzant les sentències del llenguatge incorporat al sistema gestor de bases de dades.



1.1. Procedimientos y Funciones

1.2. Procedimientos

1.2.1. Lista de parámetros de los procedimientos

1.2.2. Crear y borrar un procedimiento

1.2.3. Ejemplos de procedimiento

1.3. Funciones

1.3.1. Parámetros de una función

1.3.2. Crear y borrar una función

1.3.3. Ejemplos de una función

1.1 Procedimientos y funciones

Hay tres tipos de bloques PL/SQL:

- **Bloques Anónimos:** Estos bloques no tienen nombre y es la zona de declaraciones que comienza con la palabra reservada DECLARE.
- **Subprogramas:** Son los bloques PL/SQL que tienen un nombre y la zona de declaraciones comienza con la palabra reservada IS.

Estos subprogramas pueden ser de dos tipos:

- a)Procedimientos:** Es el tipo que más se usa en PL/SQL y normalmente se almacenan en la base de datos.
- b)Funciones:** Es otro tipo que también se usa en PL/SQL y su formato genérico es similar al de los procedimientos, pero pueden devolver un valor.

1.2 Procedimientos

```
PROCEDURE <nombre_procedimiento>  
[(<lista_de_parámetros>)]  
IS[AS]  
[<declaraciones_objetos_locales>;]  
BEGIN  
    <instrucciones>;  
[EXCEPTION  
    <excepciones>;]  
END <nombre_procedimiento>;
```

Es la cabecera, donde va el nombre del procedimiento y los parámetros.

Es el cuerpo del procedimiento y es un bloque PL/SQL. En este bloque se incluye las declaraciones, instrucciones y manejo de excepciones (si son necesarias).

1.2.1. Lista de parámetros de los procedimientos

```
PROCEDURE <nombre_procedimiento>  
[(<lista_de_parámetros>)]  
IS[AS]
```

La lista de parámetros que usa el procedimiento se ha de declarar. Si hay más de un parámetro cada uno de los parámetros separados por comas.

```
<nombre_var> [IN|OUT|IN OUT] <tipo_de_dato> [{:=|DEFAULT}<valor>]
```

nombre_var es el identificador de la variable, que ha de cumplir las reglas sobre los identificadores.

IN|OUT|IN OUT son las opciones que hacen referencia al tipo de parámetro: entrada | salida | entrada / salida.

tipo_de_dato es el tipo de datos de la variable.

valor es el valor inicial de la variable.

1.2.2. Crear y borrar un procedimiento

```
CREATE [OR REPLACE] PROCEDURE <nombre_procedimiento>
```

nombre_procedimiento, identifica el nombre del procedimiento. El nombre es un identificador por lo tanto debe ser único.

```
DROP PROCEDURE <nombre_procedimiento>
```

Una vez creado, para utilizar-lo debe ser llamado desde otro bloque de plsql.

También puede ser llamado desde un programa externo, Python, Java ... via «call».

```
[DECLARE]
```

```
BEGIN
```

```
    nombre_procedimiento(...);
```

```
END;
```

1.2.3. Ejemplos de procedimiento

Crear un procedimiento que salude a una persona por su nombre y le diga quién la saluda (por su apellido).

Los parámetros de entrada son el nombre de la persona a saludar y el apellido de la persona que saluda.

Ejemplo de procedimiento que llama a otro procedimiento:

```
CREATE OR REPLACE PROCEDURE
escribir(nombre VARCHAR2,
          apellidos VARCHAR2)
AS
BEGIN
    saludocon(UPPER(nombre));
    DBMS_OUTPUT.PUT_LINE
        ('SOY TU AMIGO, ' || UPPER(apellidos));
END escribir;
```

EJERCICIO: Llamar al procedimiento escribir desde un bloque de código anónimo

```
CREATE OR REPLACE PROCEDURE
saludocon(nombre VARCHAR2)
AS
BEGIN
    DBMS_OUTPUT.PUT_LINE
        ('HOLA SOY YO ' || nombre);
END saludocon;
```

1.3 Funciones

```
FUNCTION <nombre_funcion>
[(<lista_de_parámetros>)]
RETURN <tipo_de_valor_retorno>
IS[AS]
[<declaraciones_objetos_locales>;]
BEGIN
    <instrucciones>;
[EXCEPTION
    <excepciones>;]
END <nombre_funcion>;
```

Es la cabecera, donde va el nombre de la función, los parámetros y el valor que devuelve.

A diferencia de los procedimientos, las funciones devuelven un valor. Este valor puede ser un número, carácter, un registro, etc

1.3.1. Parámetros de una función

A la función también se la pasa parámetros como a un procedimiento, cada parámetro separado por comas.

La sintaxis es para definir un parámetro es:

`<nombre_var> [IN|OUT|IN OUT] <tipo_de_dato> [{:=|DEFAULT}<valor>]`

TIPO	DESCRIPCIÓN
IN	Es un valor de entrada. Dentro del subprograma, el parámetro actúa como una constante (no se le puede asignar ningún valor).
OUT	Es un valor de salida. Dentro del subprograma, el parámetro actúa como una variable NO inicializada.
IN OUT	Es un valor de entrada y de salida. Dentro del subprograma, el parámetro actúa como una variable inicializada.

1.3.2. Crear y borrar una función

```
CREATE [OR REPLACE] FUNCTION <nombre_funcion>
```

nombre_funcion, identifica el nombre de la función. El nombre es un identificador por lo tanto debe ser único.

```
DROP FUNCTION <nombre_funcion>
```

Una vez creada, para utilizar-la debe ser llamado desde otro bloque de plsql.

También puede ser llamada desde un programa externo, Python, Java ... via «call».

```
DECLARE  
    ret <tipo_de_valor_retorno>;  
BEGIN  
    ret:=nombre_funcion(...);  
END;
```

Preguntes!!!!

