

# Gestión de bases de datos

Procedimientos y  
funciones

# Introducción

Es un bloque PL/SQL que se compila y se almacena en la base de datos. Gracias a ellos se consigue la reutilización y por lo tanto podemos invocarlos las veces que haga falta.

Sintaxis:

```
CREATE [OR REPLACE] PROCEDURE <procedure_name>
[(<param1> [IN|OUT|IN OUT] <type>,
<param2> [IN|OUT|IN OUT] <type>,
...)]
IS -- Declaración de variables locales
BEGIN
-- Sentencias
[EXCEPTION]
-- Sentencias control de excepción
END [<procedure_name>;
```

# Parámetros

Estos pueden ser:

- ☐ **IN:** Parámetros de entrada. Es el valor por defecto en caso de que no especifiquemos nada (no haría falta indicar IN, opcional). No cambian.
- ☐ **OUT:** Parámetros de salida. Se utilizan para que el procedimiento almacene en ellos algún valor.
- ☐ **IN OUT:** Parámetros de entrada salida. Son variables cuyo valor puede ser utilizado por el procedimiento y puede almacenar un valor en ellas.

# Permisos

El usuario tiene que tener permisos para crear procedimientos en dicha base de datos. En caso de que no sea así habrá que proporcionárselos:

**grant create procedure to nombredeusuario;**

# Ejemplo

## Declaración:

```
CREATE OR REPLACE PROCEDURE Escribir(texto VARCHAR)
IS
BEGIN
    dbms_output.put_line(texto);
END;
/
```

## Invocación:

```
BEGIN
    Escribir('Groot');
End;
/
```

### Ejercicio 1

Crea un procedimiento que se llame `consultarEmpleado`. Debe tomar una variable de entrada `v_empno` con el tipo de dato del campo `empno` de la tabla `emp`. Debe tomar como variables de salida `v_ename` y `v_job`, cuyos tipos de datos deben coincidir con los de los campos `ename` y `job` de la tabla `emp`.

Controla con una excepción que no se encuentre ningún dato con el valor de `v_empno` de entrada, mostrando el mensaje “No se encontraron datos”.

### Ejercicio 2

Invoca al procedimiento consultarEmpleado pasando tres variables (id, nombre y puesto). La variable id debe obtener su valor pidiéndola por pantalla al usuario.

Se debe mostrar por pantalla el resultado devuelto del procedimiento anterior en las variables de salida nombre y puesto.

# Borrado

El comando DROP PROCEDURE seguido del nombre del procedimiento elimina el procedimiento guardado en base de datos.

Ejemplo:

```
DROP PROCEDURE consultarEmpleado;
```



### Ejercicio 3

Elimina el procedimiento consultarEmpleado y luego vuelve a ejecutar el ejercicio anterior (invocación del procedimiento con tres variables, una de entrada y dos de salida).

¿Qué sucede?

### Ejercicio 4

Escribe un procedimiento denominado `is_today` que muestre por pantalla la fecha de hoy, y luego llámalo desde un bloque anónimo.

### Ejercicio 5

Escribe un procedimiento denominado `is_today2` que, dada la fecha de hoy, la escriba por pantalla. Invoca al procedimiento con la fecha de hoy.

# Funciones

Una función es un subprograma que devuelve un valor. La sintaxis es:

```
CREATE [OR REPLACE] FUNCTION <fn_name>[(<param1> IN  
<type>, <param2> IN <type>, ...)]  
RETURN <return_type>  
IS  
    result <return_type>;  
BEGIN  
    return(result);  
[EXCEPTION]  
    -- Sentencias control de excepción  
END [<fn_name>];
```

# Funciones

El uso de REPLACE permite sobrescribir una función existente. Si se omite y la función existe, dará error.

La sintaxis de los parámetros es la misma que en los procedimientos, exceptuando que **sólo pueden ser de entrada**.

# Ejemplo

Función que calcula el cuadrado de un número.

```
CREATE OR REPLACE FUNCTION cuadrado (x NUMBER)
RETURN NUMBER
IS
BEGIN
    RETURN x*x;
END;
/
```

Llamada a la función:

```
BEGIN
    DBMS_OUTPUT.PUT_LINE(cuadrado(9));
END;
/
```

# Borrado

El comando DROP FUNCTION seguido del nombre de la función elimina la función guardada en base de datos.

Ejemplo:

```
DROP FUNCTION cuadrado;
```

### Ejercicio 6

Crear una función nombreEstudiante, que dado un código de estudiante, devuelva su nombre y apellidos. Tener en cuenta que el código no exista.



### Ejercicio 7

Crear una función `totalEstudiantes`, que devuelva el número total de estudiantes.



# CEU

*Centro de Estudios  
Profesionales*

---

*Fundación San Pablo Andalucía*