

## PROCEDIMIENTOS Y FUNCIONES ALMACENADAS – ORACLE

```
CREATE [OR REPLACE] PROCEDURE procedure_name
(argument1 [mode1] datatype1,
 argument2 [mode2] datatype2,
 ...
IS
    parte declarativa (OPCIONAL)
BEGIN
    sentencias ejecutables
[EXCEPTION]
END;
```

```
CREATE [OR REPLACE] FUNCTION FUNCTION_name
(argument1 [mode1] datatype1,
 argument2 [mode2] datatype2,
 ...
RETURN datatype
IS
BEGIN
END;
```

```
CREATE OR REPLACE PROCEDURE aumentar_salario
(v_id in emp.empno%TYPE)
IS
BEGIN
    UPDATE emp SET sal = sal * 1.10 WHERE empno = v_id;
END ;
```

Llamada a un subprograma dentro de un bloque:

Nombre\_subprograma(parámetros);

Fuera de un bloque se puede llamar usando EXECUTE Nombre\_subprograma(parámetros);

### Funciones:

Pueden ser llamadas desde bloques PL/SQL o desde expresiones SQL:

```
CREATE OR REPLACE FUNCTION get_promedio
(
    v_codigo estdiente.codigo%type
)
RETURN estudiante.promedio%TYPE
IS
    v_prom estudiante.promedio%TYPE;
BEGIN
    SELECT promedio into v_prom
```

```
from estudiante
where codigo = v_codigo;
```

```
return v_prom;
```

```
END;
```

```
select * from estudiante where promedio < get_promedio(2);
```

Parámetros:

IN	OUT	IN OUT
Por defecto	Se debe especificar	Se debe especificar
Pasa un valor al subprograma	Retorna un valor a quien llama al subprograma	Pasa un valor inicial al subprograma y retorna un valor actualizado al quien llama al subprograma
Actúa como una constante	Actúa como una variable no inicializada	Actúa como una variable inicializada
Puede ser una constante, variable inicializada, literal o una expresión.	Debe ser una variable	Debe ser una variable

No se especifica el tamaño, solamente el tipo.

Para los parámetros IN es posible asignar un valor por defecto a un parámetro usando la palabra DEFAULT seguida del valor.

```
CREATE OR REPLACE PROCEDURE aumentar_salario
(v_id in emp.empno%TYPE, v_aumento NUMBER DEFAULT 100, v_extra NUMBER DEFAULT 50)
```

Si se quiere omitir solamente el segundo parámetro en la llamada se puede hacer así:

```
Aumentar_salario(123, v_extra => 100);
```

Control de "Side Effects"

Para ejecutar una sentencia SQL que llame a una función almacenada, Oracle Server debe saber si la función es "free" o "side effects". "Side effects" son cambios a las tablas de la base de datos. Estos podrían retrasar las ejecuciones de las consultas o dar resultados dependiendo del orden (y por lo tanto, indeterminados). Por consiguiente, las restricciones se aplican a las funciones almacenadas que se llaman desde expresiones SQL.

Restricciones

- La función no puede modificar tablas de la base de datos, ni puede ejecutar una sentencia INSERT, UPDATE o DELETE.
- La función no puede llamar a otro subprograma que rompa una de las restricciones arriba indicadas..