

Índice.

1. Procedimientos.....	2
1.1. Concepto	2
1.2. Sintaxis.....	2
1.3. Ejemplo.....	2
2. Funciones.....	3
2.1. Concepto	3
2.2. Sintaxis.....	3
2.3. Ejemplo.....	3
3. Disparadores.....	4
3.1. Concepto	4
3.2. Sintaxis.....	4

1. Procedimientos.

1.1. Concepto.

Un **procedimiento** es un código de PL/SQL que se encarga de realizar una determinada tarea.

1.2. Sintaxis.

La sintaxis de un procedimiento es la siguiente:

```
CREATE {OR REPLACE} PROCEDURE nombre ( [IN | OUT | INOUT] nombre_parametro tipo_variable, ... )  
    [DECLARE]  
        -- declaración de variables locales  
    BEGIN  
        -- instrucciones de ejecución  
    [EXCEPTION]  
        -- instrucciones de excepción  
    END;
```

1.3. Ejemplo.

```
DELIMITER //  
CREATE PROCEDURE factorial_pro(IN numero int, OUT factorial int8)  
    BEGIN  
        SET factorial = 1;  
        WHILE numero > 1 DO  
            SET factorial := factorial * numero;  
            SET numero := numero - 1;  
        END WHILE;  
    END  
// DELIMITER ;
```

2. Funciones.

2.1. Concepto.

Un **función (o function)** es un código de PL/SQL similar a un procedimiento, pero con la diferencia de que devuelve un valor de un tipo concreto.

2.2. Sintaxis.

La sintaxis de una función es la siguiente:

```
CREATE {OR REPLACE} FUNCTION nombre ( nombre_parametro tipo_variable, ... )  
    RETURNS tipo  
    BEGIN  
        -- instrucciones de ejecución  
    [EXCEPTION]  
        -- instrucciones de excepción  
    END
```

2.3. Ejemplo.

```
DELIMITER //  
CREATE FUNCTION factorial(numero int) RETURNS INT8  
    DETERMINISTIC  
    BEGIN  
        DECLARE fact int8;  
        SET fact := 1;  
        WHILE numero > 1 DO  
            SET fact := fact * numero;  
            SET numero := numero - 1;  
        END WHILE;  
        RETURN fact;  
    END  
// DELIMITER ;
```

3. Disparadores.

3.1. Concepto.

Un **trigger** (o disparador) es un código de PL/SQL similar a un procedimiento (o función), pero con la peculiaridad de estar asociado a una tabla y que se ejecuta automáticamente como reacción a una operación DML específica (INSERT, UPDATE o DELETE).

3.2. Sintaxis.

La sintaxis de un trigger es la siguiente:

```
CREATE {OR REPLACE} TRIGGER nombre
  [BEFORE | AFTER] [DELETE | INSERT | UPDATE {OF columnas}] ON table
  [FOR EACH ROW [WHEN condicion]]
  [DECLARE]
    -- declaración de variables locales
  BEGIN
    -- instrucciones de ejecución
  [EXCEPTION]
    -- instrucciones de excepción
  END;
```

[BEFORE | AFTER] indica cuándo se ejecutará el código si antes o después de la sentencia DML
 [DELETE | INSERT | UPDATE] operación DML involucrada. Se puede elegir una columna

orden	old	new
INSERT	No definido. Todo NULL.	Nuevos valores.
UPDATE	Valores antes de actualización.	Valores tras la actualización.
DELETE	Valores antes del borrado.	No definido. Todo NULL

[FOR EACH ROW] el trigger se dispara cada vez que se realiza una operación en una fila (row)
 [WHEN] el trigger sólo actuará sobre las filas que cumplan la restricción
 [DECLARE] declaración de variables dentro del trigger
 BEGIN cuerpo del programa, puede llamar a un procedimiento o función
 [EXCEPTION] tratamiento de errores o excepciones