

SINTAXIS VISTAS, PROCEDIMIENTOS, FUNCIONES y TRIGGERS

Vistas:

CREATE OR REPLACE VIEW nombre-vista [(columnas)] AS

SELECT

Ejecución:

SELECT * FROM nombre-vista;

Procedimientos:

DROP PROCEDURE IF EXISTS nombre-procedimiento;

DELIMITER \$\$ (símbolo a escoger)

CREATE PROCEDURE nombre-procedimiento ([parametros])

BEGIN

 [declaración e inicialización de variables;]

 instrucciones;

END

\$\$

Parametros:

IN|OUT|INOUT parametro TIPODATO[, ...] #NOTA: el valor por defecto es IN (de entrada)

Declaración e inicialización de variables:

DECLARE variables [, ...] TIPODATO [DEFAULT valor];

Asignación de variables:

SET variable = valor[, ...];

Comentarios, 3 posibilidades:

comentario

-- comentario

/* comentarios */

Llamada al procedimiento:

CALL nombre-procedimiento([parámetros]);

Funciones:

```
/*SET GLOBAL log_bin_trust_function_creators = 1;*/
```

```
DROP FUNCTION IF EXISTS nombre-funcion;
```

```
DELIMITER $$
```

```
CREATE FUNCTION nombre-función([parametros]) RETURNS TIPODATO
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    [declaración e inicialización de variables;]
```

```
    instrucciones;
```

```
    RETURN variable;
```

```
END
```

```
$$
```

Llamada a la función(NOTA: se le puede invocar o llamar desde cualquier “sitio”):

- SELECT funcion();
- SELECT CODIGO, DNI, NOMBRE, función(),

- SET var = funcion();
- ...

- IF funcion() = 0
- ...

Triggers:

```
DROP TRIGGER IF EXISTS nombre-trigger;
```

```
DELIMITER $$
```

```
CREATE TRIGGER nombre-trigger momento evento
```

```
ON tabla FOR EACH ROW
```

```
BEGIN
```

```
    [declaración e inicialización de variables;]
```

```
    instrucciones ;
```

```
END
```

```
$$
```

momento:

BEFORE | **AFTER**

Evento:

INSERT | **UPDATE** | **DELETE**

Referenciar atributos de la propia tabla: old.atributo Ó new.atributo

CURSORES:

NOTA: Un CURSOR se puede declarar dentro de un procedimiento, función o de un trigger. El objetivo es obtener una serie de registros con 1 o varios campos para poder después “operar” individualmente con cada uno de ellos.

Pasos:

- 1) Declarar el cursor
- 2) Declarar un handler(para poder continuar con el siguiente registro hasta llegar al último)
- 3) Abrir el cursor
- 4) Realizar el “volcado” de la(s) variable(s) mediante el FETCH dentro de un WHILE o REPEAT
- 5) Cerrar el cursor.

BEGIN

declaración e inicialización de variables

DECLARE FINAL INT DEFAULT 0;

DECLARE nombrecursor CURSOR FOR #declaración del cursor

SELECT

DECLARE CONTINUE HANDLER FOR NOT FOUND SET FINAL = 1; #declaración de un handler(manejador)

##más opciones válidas

/*DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET FINAL = 1;*/

/*DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET FINAL = 1;*/

/*DECLARE CONTINUE HANDLER FOR 1329 SET FINAL = 1;*/

OPEN nombrecursor; #apertura del cursor

WHILE NOT FINAL DO #FINAL: variable de tipo entero

FETCH nombrecursor INTO variable(s); /*volcado(asignación) de los valores del cursor en una o varias
variables, en función de los campos obtenidos en la SELECT de la declaración*/

Instrucciones; #pueden ser de distinto tipo, incluyendo UPDATE, INSERT, DELETE,...

END WHILE;

CLOSE nombrecursor;

END