

# Procedimientos almacenados, triggers, funciones y vistas.

## Procedimientos almacenados

Procedimientos almacenados: conjunto de instrucciones SQL junto a una serie de estructuras que permiten dar lógica a algún procedimiento. Los procedimientos se almacenan en el servidor y pueden llamarse para su acceso.

Sintaxis

Crear → CREATE PROCEDURE

Llamar → CALL

--

Estructura:

- ✓ Nombre
- ✓ Parámetros
- ✓ Contenido
- ✓ Acción
- ✓ Instrucciones, variables, etc.

---

Ejemplos

-- crear base de datos especificando nombre

DELIMITER \$\$

```
CREATE PROCEDURE crear_bd (IN bd VARCHAR(60))
BEGIN
```

```
SET @nombre_bd = bd;
SET @cadena = CONCAT('create database ',@nombre_bd);
PREPARE cadena FROM @cadena;
EXECUTE cadena;
```

END \$\$

DELIMITER ;

-- crear tabla especificando nombre

DELIMITER \$\$

```
CREATE PROCEDURE crear_tabla(IN tabla VARCHAR(60))
BEGIN
```

```
SET @nombre_tabla = tabla;
SET @cadena = CONCAT('CREATE TABLE ',@nombre_tabla,'(id int(11) default NULL, nombre varchar(100) default NULL)');
```

```
PREPARE cadena FROM @cadena;  
EXECUTE cadena;  
END $$  
DELIMITER ;
```

-- crear sp con acción de insertar en una tabla especificada

```
DELIMITER $$  
CREATE PROCEDURE sp_insertar(IN id INT, nombre varchar(60))  
BEGIN  
    Insert into tbl1 values(id,nombre);  
END $$  
DELIMITER ;
```

-- crear tabla llamada acceso que servirá más adelante

```
DELIMITER $$  
CREATE PROCEDURE crear_acceso(IN tabla VARCHAR(60))  
BEGIN  
SET @nombre_tabla = tabla;  
SET @cadena = CONCAT('CREATE TABLE ',@nombre_tabla,'(id int(11) default NULL, nombre varchar(100) default NULL,  
acceso DateTime)');  
PREPARE cadena FROM @cadena;  
EXECUTE cadena;  
END $$  
DELIMITER ;
```

-- crear un procedimiento almacenado con un Case

```
DELIMITER $$  
CREATE PROCEDURE sp_consultas(IN Numero INT)  
BEGIN  
SET @num = Numero;  
    case @num  
        when 1 then select nombre from tbl1;  
        when 2 then select * from tbl1;
```

```
        else
            select user();
        end case;
END $$
DELIMITER ;
```

```
*/
```

Para borrar los procedimientos usamos drop procedure

```
*/
```

Borrar – Drop procedure <Nombre>

## Triggers

Disparadores: objetos dentro de una base de datos asociado a una tabla, se activa al ocurrir un evento específico.

```
DELIMITER $$
CREATE TRIGGER ultimo_acceso
BEFORE INSERT ON tbl1 for each row
BEGIN
INSERT INTO acceso(id, nombre, acceso) values(1,'admin', current_timestamp());
END $$
DELIMITER ;
```