



Desarrollo de Sistemas Web

Bases de Datos

FES Aragón

MTI. Omar Mendoza González

Stored Procedures en MySQL

- Son Rápidos
- No son compilados
- Reducen el Trafico en la Red
- Se pueden utilizar para programar funciones que requieren de un gran numero de chequeos, búsquedas y sentencias sin la necesidad de la participación interactiva del usuario durante varios pasos de una misma tarea.
- Pueden ser considerados como un componente
- Son potables
- Siempre esta disponible el código fuente

Como se Crea un SP

- Sentencia de Creación
- Nombre de Procedimiento
- Lista de Parámetros
- Cuerpo del Procedimiento

- Sentencias Admitidas en un SP
 - INSERT, UPDATE, DELETE, SELECT, DROP, CREATE, REPLACE

Sintaxis

```
CREATE PROCEDURE nombre_sp ([IN] | [OUT] | [INOUT])  
LANGUAGE SQL  
NOT DETERMINISTIC  
SQL SECURITY DEFINER | INVOKER  
COMMENT 'Esto es un Procedimiento Almacenado'  
SELECT CURRENT_DATE, RAND() FROM empleado //
```

SP de ejemplo

```
CREATE PROCEDURE lista_empleados()  
    SELECT * FROM empleado; //
```

```
CREATE PROCEDURE p ()  
    DELETE FROM t; //
```

```
CREATE PROCEDURE p1 ()  
    SET @x = 5; //
```

Ejemplo de SP con Parámetro IN

```
DELIMITER //  
CREATE PROCEDURE sp1(p INT) SET @x = p //  
DELIMITER ;
```

```
CALL sp1(12345);
```

```
SELECT @x;
```

```
+-----+  
| @x    |  
+-----+  
| 12345 |  
+-----+
```

Ejemplo de SP con Parámetro OUT

```
CREATE PROCEDURE sp2 (OUT p INT)  
SET p = -5 //
```

```
CALL sp2 (@y)//
```

```
SELECT @y//
```

Condiciones IF-THEN-ELSE

```
CREATE PROCEDURE sp3 (IN parameter1 INT)
BEGIN
    DECLARE variable1 INT;
    SET variable1 = parameter1 + 1;
    IF variable1 = 0 THEN
        INSERT INTO t VALUES (17);
    END IF;
    IF parameter1 = 0 THEN
        UPDATE t SET valor = valor + 1;
    ELSE
        UPDATE t SET valor = valor + 2;
    END IF;
END; //
```


Condiciones CASE

```
CREATE PROCEDURE sp4 (IN parameter1 INT)
  BEGIN
    DECLARE variable1 INT;
    SET variable1 = parameter1 + 1;
    CASE variable1
      WHEN 0 THEN INSERT INTO t VALUES (17);
      WHEN 1 THEN INSERT INTO t VALUES (18);
      ELSE INSERT INTO t VALUES (parameter1);
    END CASE;
  END; //
```

Secuencias WHILE-END WHILE

```
CREATE PROCEDURE sp5 (IN parameter1 INT)
  BEGIN
    DECLARE v INT;
    SET v = 0;
    WHILE v < parameter1 DO
      INSERT INTO t VALUES (v);
      SET v = v + 1;
    END WHILE;
  END; //
```

Secuencias REPEAT-END REPEAT

```
CREATE PROCEDURE sp6 ()  
  BEGIN  
    DECLARE v INT;  
    SET v = 0;  
    REPEAT  
      DELETE FROM t WHERE valor = v;  
      SET v = v + 1;  
    UNTIL v >= 5  
    END REPEAT;  
END; //
```

Secuencias LOOP-END LOOP

```
CREATE PROCEDURE sp7 ()  
  BEGIN  
    DECLARE v INT;  
    SET v = 0;  
    loop_label: LOOP  
      INSERT INTO t VALUES (v);  
      SET v = v + 1;  
      IF v >= 5 THEN  
        LEAVE loop_label;  
      END IF;  
    END LOOP;  
  END;  
//
```

LABEL e INTERATE

```
CREATE PROCEDURE sp8 ()  
  BEGIN  
    DECLARE v INT;  
    SET v = 0;  
    loop_label: LOOP  
      IF v = 3 THEN  
        SET v = v + 1;  
        ITERATE loop_label;  
      END IF;  
      INSERT INTO t VALUES (v);  
      SET v = v + 1;  
      IF v >= 5 THEN  
        LEAVE loop_label;  
      END IF;  
    END LOOP;  
  END;
```

Cursors

```
DECLARE cursor-name CURSOR FOR SELECT ...;  
OPEN cursor-name;  
FETCH cursor-name INTO variable [, variable];  
CLOSE cursor-name;
```

Características de los Cursores

- READ ONLY
- NOT SCROLLABLE
- ASENSITIVE

Ejemplo de Cursores

```
CREATE PROCEDURE sp9 (OUT return_val INT)
BEGIN
  DECLARE a,b INT;
  DECLARE cur_1 CURSOR FOR SELECT valor FROM t;
  DECLARE CONTINUE HANDLER FOR NOT FOUND
  SET b = 1;
  OPEN cur_1;
  REPEAT
    FETCH cur_1 INTO a;
  UNTIL b = 1
  END REPEAT;
  CLOSE cur_1;
  SET return_val = a;
END;
```

Orden de declaración

- El orden de declaración es importante:
 1. Variables
 2. Condiciones.
 3. Cursors.
 4. Handlers

Insertar Información

```
CREATE PROCEDURE add_empleado
(id_emp INT, nombre_emp VARCHAR(30), paterno_emp VARCHAR(30), materno_emp
  VARCHAR(30))
SQL SECURITY DEFINER
BEGIN
  IF id_emp < 0 OR id_emp = " OR nombre_emp=" OR nombre_emp=" OR paterno_emp="
    OR materno_emp=" THEN
    SELECT 'Warning: Parametros Invalidos'AS error;
  END IF;
  INSERT INTO empleado (empno,enombre,epaterno,ematerno)
  VALUES (id_emp,nombre_emp,paterno_emp,materno_emp);
  SELECT 'Empleado Insertado';
  INSERT INTO info_empleado (empno) VALUES (id_emp);
  SELECT 'Info empleadao Insertado';
END;
```

Respaldar Información

```
CREATE PROCEDURE resp_dpto ()
BEGIN
    DECLARE id_d, b,a INT;
    DECLARE nom_d VARCHAR(15);
    DECLARE loc_d VARCHAR(20);
    DECLARE cur_1 CURSOR FOR SELECT * FROM depto;
    DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET b = 1;
    OPEN cur_1;
    SET a = 1;
    REPEAT
        FETCH cur_1 INTO id_d,nom_d,loc_d;
        INSERT INTO depto_res VALUES (id_d,nom_d,loc_d);
        SET a = a+1;
    UNTIL b = 1
    END REPEAT;
    CLOSE cur_1;
    SELECT CONCAT(a, ' Registros respaldados') AS resultado;
END;
```