



El empleo
es de todos

Mintrabajo

DESARROLLO DE BASE DE DATOS CON MYSQL



@SENAcomunica

www.sena.edu.co



DESARROLLO DE BASE DE DATOS CON MYSQL



@SENAcomunica

www.sena.edu.co

13

PROCEDIMIENTOS ALMACENADOS

OBJETIVOS



Después de completar esta lección usted estará en la capacidad de:

- **Identificar los beneficios del diseño de programas modularizados.**
- **Crear y llamar procedimientos.**
- **Usar parámetros en los procedimientos**
- **Usar parámetros con notación posicional, nombrados o mixtos en los procedimientos.**
- **Identificar modos de pasar valores por parámetros**
- **Borrar procedimientos.**

SINTAXIS PROCEDIMIENTOS Y FUNCIONES



```
CREATE PROCEDURE sp_name ([parameter[,...]])  
    [characteristic ...] routine_body  
  
CREATE FUNCTION sp_name ([parameter[,...]])  
    RETURNS type  
    [characteristic ...] routine_body  
  
parameter:  
    [ IN | OUT | INOUT ] param_name type  
  
type:  
    Any valid MySQL data type  
  
characteristic:  
    LANGUAGE SQL  
    | [NOT] DETERMINISTIC  
    | { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }  
    | SQL SECURITY { DEFINER | INVOKER }  
    | COMMENT 'string'  
  
routine_body:  
    procedimientos almacenados o comandos SQL válidos
```

Los beneficios de utilizar constructores de programación modular son:

- **Fácil mantenimiento.**
- **Mejoramiento de la seguridad y la integridad.**
- **Mejoramiento de su ejecución.**
- **Mejoramiento en la claridad del código.**

QUÉ SON LOS SUBPROGRAMAS



- **Un subprograma es un bloque nombrado que puede ser llamados con un conjunto de parámetros.**
- **Un subprograma consiste en una especificación y un cuerpo.**
- **Un subprograma puede ser un procedimiento o una función.**
- **Un procedimiento (PROCEDURE), se usa para ejecutar una acción y una función (FUNCTION), para retornar un valor.**

QUÉ SON LOS PROCEDIMIENTOS



A partir de mysql 5.0, es posible definir rutinas en MySQL. Éstas son funciones y procedimientos que están almacenadas en el servidor de bases de datos y que pueden ser ejecutadas posteriormente. Las funciones (stored functions) regresan un resultado y pueden ser utilizadas en expresiones (de la misma manera en que se usan las funciones de MySQL).

QUÉ SON LOS PROCEDIMIENTOS



- **Son un tipo de subprograma que ejecuta una acción.**
- **Se almacenan en la base de datos.**
- **Promueve la reusabilidad y el mantenimiento.**

Las ventajas del uso de las rutinas anteriores son las siguientes:

- **Extienden la sintaxis de SQL al agregar ciclos e instrucciones de saltos.**
- **Proveen un mecanismo de manejo de errores.**
- **Debido a que están almacenadas en el servidor, todo el código necesario para definirlas necesita ser mandado por la red solamente una vez, en el momento de su creación y no cada vez que es invocada. Lo anterior reduce sobrecarga.**
- **Los procedimientos almacenados pueden mejorar el rendimiento ya que se necesita enviar menos información entre el servidor y el cliente.**
- **Los procedimientos almacenados le permiten tener bibliotecas o funciones en el servidor de base de datos.**

Porque es importante el uso del DELIMITER: por defecto mysql usa como DELIMITER el punto y coma (;) , es decir, cada vez que encuentre punto y coma(;) ejecuta hasta ahí, debido a que los procedimientos y funciones son varias líneas de códigos y algunas de ellas terminan con este delimiter se ejecutaría solo hasta ahí, lo que ocasionaría un error, es por esto que se hace necesario indicarle a mysql que utilice otro DELIMITER que puede ser cualquiera para el ejemplo usamos \$ y al finalizar la creación del procedimiento o función volvemos a cambiarlo por (;)

CREANDO PROCEDIMIENTOS CON CREATE O REPLACE



- Use la cláusula **CREATE** para crear un procedimiento.
- Use la opción **OR REPLACE** para sobre escribir un procedimiento existente.

```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE PR_PRUEBA()
-> BEGIN
->     DECLARE X INTEGER;
-> END;
-> $
Query OK, 0 rows affected (0.02 sec)

MariaDB [HR]> DELIMITER ;
```

QUÉ SON LOS PARAMETROS Y LOS MODOS



- Son declarados después de el nombre del subprograma.
- Permite pasar datos al subprograma.
- Se usa desde variable locales pero dependen del modo como se pase el parámetro, así:
- De manera predeterminada, un parámetro de un procedimiento es de tipo IN; un parámetro definido de esta manera se recibe en el procedimiento pero cualquier modificación realizada en él no se conservará una vez que el procedimiento termine. Un parámetro OUT es lo contrario: el procedimiento asignará algún valor al parámetro, el cual podrá ser accedido una vez que el procedimiento haya regresado. Un parámetro INOUT permite mandar un valor al procedimiento y obtenerlo de vuelta.

LOS PARÁMETROS Y LOS MODOS



IN	OUT	INOUT
Modo por defecto	Debe ser especificado	Debe ser especificado
Valores son pasados al subprograma	Return valores hacia el exterior	Return valores hacia el exterior
Parámetros son pasado como una constante	Variable sin inicializar	debe inicializarse la variable
Un parámetro puede ser una literal, expresión, constante o variable	Debe ser una variable	Debe ser una variable
Puede ser asignado como un valor por defecto	No se puede asignar con un valor por defecto	No se puede asignar con un valor por defecto

LOS PARÁMETROS Y LOS MODOS (IN)



```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE ACTUALIZAR_SALARIO(IN PID INTEGER,IN INCREMENTO INTEGER)
-> BEGIN
->   UPDATE EMPLOYEES SET SALARY=SALARY*(1+INCREMENTO/100)
->   WHERE EMPLOYEE_ID=PID;
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]> SELECT EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) NOMBRE,
->   SALARY SALARIO
->   FROM EMPLOYEES WHERE EMPLOYEE_ID=101;
+-----+-----+-----+
| EMPLOYEE_ID | NOMBRE          | SALARIO |
+-----+-----+-----+
|          101 | Kochhar Neena   | 17000.00 |
+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [HR]>
MariaDB [HR]> CALL ACTUALIZAR_SALARIO(101,5);
Query OK, 1 row affected (0.02 sec)

MariaDB [HR]>
MariaDB [HR]> SELECT EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) NOMBRE,
->   SALARY SALARIO
->   FROM EMPLOYEES WHERE EMPLOYEE_ID=101;
+-----+-----+-----+
| EMPLOYEE_ID | NOMBRE          | SALARIO |
+-----+-----+-----+
|          101 | Kochhar Neena   | 17850.00 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

LOS PARÁMETROS Y LOS MODOS (MAL USO DE OUT)



```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE ACTUALIZAR_SALARIO(IN PID INTEGER,IN INCREMENTO INTEGER, OUT IVA INTEGER)
-> BEGIN
->   SET IVA=8;
->   UPDATE EMPLOYEES SET SALARY=SALARY*(1+INCREMENTO/100)
->   WHERE EMPLOYEE_ID=PID;
-> END;
-> $
Query OK, 0 rows affected (0.02 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]> CALL ACTUALIZAR_SALARIO(101,5,16);
ERROR 1414 (42000): OUT or INOUT argument 3 for routine hr.ACTUALIZAR_SALARIO is not a variable or NEW pseudo-variable in BEFORE trigger
```


LOS PARÁMETROS Y LOS MODOS (OUT)



```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE CALCULE_IVA(IN PID INTEGER,OUT SALARIO INTEGER)
-> BEGIN
->     SELECT SALARY INTO SALARIO FROM EMPLOYEES
->     WHERE EMPLOYEE_ID=PID;
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE PROYECTE_SALARIO(IN PID INTEGER)
-> BEGIN
->     DECLARE X INTEGER;
->     CALL CALCULE_IVA(PID,X);
->     SELECT X*0.86;
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]> CALL PROYECTE_SALARIO(101);
+-----+
| x*0.86 |
+-----+
| 15351.00 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)
```

LOS PARÁMETROS Y LOS MODOS (INOUT)



```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE CALCULE_IVA(IN PID INTEGER,INOUT SALARIO INTEGER,IN IVA INTEGER)
-> BEGIN
->     SELECT SALARIO INTO SALARIO FROM EMPLOYEES
->     WHERE EMPLOYEE_ID=PID;
->     SELECT SALARIO*(1+IVA/100) INTO SALARIO;
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE PROYECTE_SALARIO(IN PID INTEGER)
-> BEGIN
->     DECLARE X INTEGER;
->     CALL CALCULE_IVA(PID,X,16);
->     SELECT X;
-> END;
-> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]> CALL PROYECTE_SALARIO(101);
+-----+
| X      |
+-----+
| 20706  |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.04 sec)
```



G R A C I A S

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co

CRÉDITOS



Realizado por el instructor José Fernando Galindo Suárez
jgalindos@sena.edu.co 2020

