



El empleo
es de todos

Mintrabajo

SQL PARA MYSQL



@SENAcomunica

www.sena.edu.co



PROCEDIMIENTOS ALMACENADOS EN MYSQL

Instructor: José Fernando Galindo Suárez
jgalindos@sena.edu.co 17003 CGMLTI



MySQL®

Mysql

Es una base de datos desarrollado inicialmente por MYSQL AB, fundada por David Axmark, Alan Larsson y Michael Widenius, luego en el 2008 fue adquirido por Sun Microsystems y en el 2010 por Oracle Corporation.

La comunidad que ayudo a desarrollar mysql, desarrollaron un nuevo gestor de base de datos llamado MariaDB.



Objetivos



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

- Identificar las ventajas del diseño de subprogramas modularizados
- Crear y llamar procedimientos
- Usar parámetros.
- Pasar datos a parámetros con notación normal, posicionado y combinados
- Identificar los diferentes modos de pasar información por valor y por referencia.
- Manejar excepciones en los procedimientos
- Borrar un procedimiento
- Desplegar información de los procedimientos

Transacciones



Las transacciones son un conjunto de instrucciones SQL que tienen la cualidad de ejecutarse como una unidad, o se ejecutan todas o no se ejecuta ninguna. Si una transacción tiene éxito, todas las modificaciones de los datos realizados durante la transacción se guardan en la base de datos. Si una transacción contiene errores los cambios no se guardarán en la base de datos.

El ejemplo clásico de transacción es una transferencia bancaria, en la que quitamos saldo a una cuenta y lo añadimos en otra.

Transacciones



Un procedimiento almacenado es un conjunto de comandos SQL que pueden almacenarse en el servidor.

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.

Funciones y 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).

Procedimientos



Los procedimientos (stored procedures) no regresan ningún valor directamente. Sin embargo, soportan tipos de parámetros cuyo valor puede fijarse en el interior del procedimiento, de tal manera que dicho valor pueda ser utilizado una vez que el procedimiento haya terminado.

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 definir las 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.

Parámetros



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.

DELIMITER



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 ;

Manejadores de errores (HANDLERS)



Los handlers de MySQL permiten hacer un tratamiento específico de ciertas condiciones (estas condiciones suelen estar relacionadas con errores).

Para un handler `CONTINUE` , continúa la rutina actual tras la ejecución del comando del handler.

Para un handler `EXIT` , termina la ejecución del comando compuesto `BEGIN...END` actual.

El handler de tipo `UNDO` todavía no se soporta.

Manejadores de errores (HANDLERS)



SQLWARNING es una abreviación para todos los códigos SQLSTATE que comienzan con 01.

NOT FOUND es una abreviación para todos los códigos SQLSTATE que comienzan con 02.

SQLEXCEPTION es una abreviación para todos los códigos SQLSTATE no tratados por SQL-WARNING o NOT FOUND.

Excepciones



El uso de las excepciones, procedimientos almacenados, transacciones y handlers en base de datos, nos es de gran ayuda ya que, gracias a ellos, tenemos una mayor seguridad y mayores rendimientos en el servidor ya que existe menos comunicación entre el cliente y el servidor.

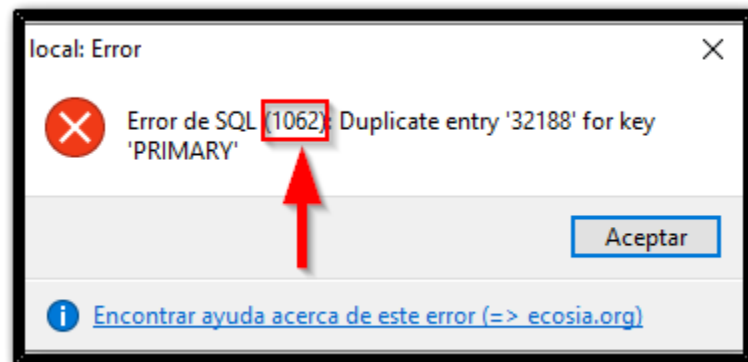
Manejadores de errores (HANDLERS)



Además de los valores SQLSTATE , los códigos de error MySQL se soportan, por ejemplo:

```
delimiter $
CREATE PROCEDURE `SP_USUARIO_CREA` (
IN `PID` INT,
IN `PNOMBRE` VARCHAR(50),
IN `PAPELLIDO` VARCHAR(50),
IN `PTIPO` INT,
IN `PSEXO` CHAR(1))
BEGIN
INSERT INTO
PERSONA (IDPERSONA, NOMBRE, APELLIDO, TIPOIDENTIFICACION
, SEXO) VALUES (PID, PNOMBRE, PAPELLIDO, PTIPO, PSEXO);
END
$
```

```
CALL
SP_USUARIO_CREA(32188, 'CARLOS', 'MARTINEZ', 1, 'M')
```




Manejadores de errores (HANDLERS)



Además de los valores SQLSTATE , los códigos de error MySQL se soportan, por ejemplo:

```
1 delimiter $
2 CREATE PROCEDURE `SP_USUARIO_CREA` (
3   IN `PID` INT,
4   IN `PNOMBRE` VARCHAR(50),
5   IN `PAPELLIDO` VARCHAR(50),
6   IN `PTIPO` INT,
7   IN `PSEXO` CHAR(1))
8 BEGIN
9   DECLARE CONTINUE HANDLER FOR 1062
10  BEGIN
11    SELECT 'EL ID YA EXISTE' AS MSG;
12  END;
13 INSERT INTO PERSONA(IDPERSONA,NOMBRE,APELLIDO,TIPOIDENTIFICACION,SEXO) VALUES(PID,PNOMBRE,PAPELLIDO,PTIPO,PSEXO);
14 END
15 $
```



```
CALL
SP_USUARIO_CREA (32188, 'CARLOS', 'MARTINEZ'
,1, 'M');
```

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0000 segundos.)

CALL SP_USUARIO_CREA(32188,'CARLOS','MARTINEZ',1,'M')

[Editar en línea] [Editar] [Crear código PHP]

☐ Mostrar todo | Número de filas: 25 ▼ | Filtrar filas:

+ Opciones

MSG
EL ID YA EXISTE



Resultado #1 (1×1)	
MSG	
EL ID YA EXISTE	

Manejadores de errores (HANDLERS)



```
delimiter $
CREATE PROCEDURE `SP_USUARIO_ACTUALIZA` (
  INOUT `RESULTADO` VARCHAR(50),
  IN `PID` INT,
  IN `PNOMBRE` VARCHAR(50),
  IN `PAPELLIDO` VARCHAR(50),
  IN `PTIPO` INT,
  IN `PSEXO` CHAR(1)
)
BEGIN
  SET @X=(SELECT COUNT(*) FROM PERSONA WHERE IDPERSONA=PID);

  IF(@X = 0) THEN
    SET RESULTADO=('**ERROR: EL ID NO EXISTE');
  ELSE
    UPDATE PERSONA SET NOMBRE=PNOMBRE, APELLIDO=PAPELLIDO, TIPOIDENTIFICACION=PTIPO,
    SEXO=PSEXO WHERE IDPERSONA=PID ;
    SET RESULTADO=('ACTUALIZADO CORRECTAMENTE');
  END IF;
END
$
```

Manejadores de errores (HANDLERS)



```
CALL SP_USUARIO_ACTUALIZA (@R, 321881, 'LUIS', 'SANCHEZ', 1, 'M');  
SELECT @R;
```

Mostrar ventana de consultas SQL

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas) (La consulta tardó 0,0000 segundos.)

CALL SP_USUARIO_ACTUALIZA(@R, 321881, 'LUIS', 'SANCHEZ', 1, 'M')

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles. ⓘ

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0000 segundos.)

SELECT @R

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo

Número de filas:

25

Filtrar filas:

Buscar en esta tabla

+ Opciones

@R

**ERROR: EL ID NO EXISTE

Manejadores de errores (HANDLERS)



```
CALL SP_USUARIO_ACTUALIZA(@R,32188,'LUIS','SANCHEZ',1,'M');  
SELECT @R;
```

Mostrar ventana de consultas SQL

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0000 segundos.)

CALL SP_USUARIO_ACTUALIZA(@R,32188,'LUIS','SANCHEZ',1,'M')

[Editar en línea] [Editar] [Crear código PHP]

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles. ⓘ

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0000 segundos.)

SELECT @R

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 ▾ Filtrar filas:

+ Opciones

@R
ACTUALIZADO CORRECTAMENTE

Manejadores de errores (HANDLERS)



```
delimiter $
CREATE PROCEDURE `SP_USUARIO_BORRA` (
  INOUT `RESULTADO` VARCHAR(50),
  IN `PID` INT,
  IN `PNOMBRE` VARCHAR(50),
  IN `PAPELLIDO` VARCHAR(50),
  IN `PTIPO` INT,
  IN `PSEXO` CHAR(1)
)
BEGIN
  SET @X=(SELECT COUNT(*) FROM PERSONA WHERE IDPERSONA=PID);

  IF(@X = 0) THEN
    SET RESULTADO=('**ERROR: EL ID NO EXISTE');
  ELSE
    DELETE FROM PERSONA WHERE IDPERSONA=PID ;
    SET RESULTADO=('BORRADO CORRECTAMENTE');
  END IF;
END
$
```

Manejadores de errores (HANDLERS)



```
CALL SP_USUARIO_BORRA(@R,32188,'LUIS','SANCHEZ',1,'M');  
SELECT @R;
```

Mostrar ventana de consultas SQL

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0000 segundos.)

CALL SP_USUARIO_BORRA(@R,32188,'LUIS','SANCHEZ',1,'M')

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles. ⓘ

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0000 segundos.)

SELECT @R

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 ▾ Filtrar filas:

+ Opciones

@R
BORRADO CORRECTAMENTE



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



Atribución, no comercial, compartir igual

Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.



Creative Commons