



Khoa CNTT
Đại Học Đà Lạt



HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

MYSQL STORED PROCEDURE

Th.s. Đoàn Minh Khuê
khuedm@dlu.edu.vn

Định nghĩa

Tạo MySQL Stored Procedure

Biến

Truyền tham số

Câu lệnh IF ELSE

Câu lệnh CASE

Vòng lặp While



Định nghĩa

- ❖ Tạo những hàm (Procedure) để thực hiện những dòng lệnh liên quan trong đó, ví dụ như thao tác Update hay Insert.
- ❖ Thông thường một hàm có thể gọi lại chính nó và ta gọi là đệ quy



Ưu điểm

- ❖ Giảm thời gian giao tiếp giữa các ứng dụng với hệ quản trị MYSQL
- ❖ Giúp các ứng dụng nhìn minh bạch hơn



Nhược điểm

- ❖ Tạo ra quá nhiều Procedure thì hệ quản trị sẽ sử dụng khá nhiều bộ nhớ để lưu trữ các thủ tục này.
- ❖ Thực hiện quá nhiều xử lý trong mỗi thủ tục thì sẽ bắt CPU làm việc nặng hơn

Tạo Stored Procedure

❖ Cú pháp:

```
DELIMITER $$  
CREATE PROCEDURE procedureName()  
BEGIN  
    /*Xu ly*/  
END; $$  
DELIMITER ;
```

❖ Trong đó:

- **DELIMITER \$\$** dùng để phân cách bộ nhớ lưu trữ thủ tục Cache và mở ra một ô lưu trữ mới.
- **CREATE PROCEDURE** dùng để khai báo tạo một Procedure mới.
- **procedureName()**: tên của thủ tục
- **BEGIN** và **END; \$\$** dùng để khai báo bắt đầu của Procedure và kết thúc Procedure
- **DELIMITER**: đóng lại ô lưu trữ

Tạo Stored Procedure

- ❖ Ví dụ: Tạo một **Procedure** với tên là **GetAllProducts()**, để truy vấn lấy danh sách tất cả sản phẩm nằm trong bảng **Products**.

```
CREATE TABLE IF NOT EXISTS `products` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `title` VARCHAR(255) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `content` TEXT COLLATE utf8_unicode_ci,  
  PRIMARY KEY (`id`)  
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=3 ;
```

```
INSERT INTO `products` (`id`, `title`, `content`) VALUES  
(1, 'he quan tri csdl', 'Cach tao va su dung stored Procedure'),  
(2, 'Tutorials học Stored Procedure', 'Website Tutorials học  
Stored Procedure');
```



Tạo Stored Procedure

- ❖ Ví dụ: Tạo một **Procedure** với tên là **GetAllProducts()**, để truy vấn lấy danh sách tất cả sản phẩm nằm trong bảng **Products**.

```
DELIMITER $$  
CREATE PROCEDURE GetAllProducts()  
BEGIN  
    /*Xu ly*/  
END; $$  
DELIMITER ;
```


Gọi Stored Procedure

❖ Cú pháp:

```
CALL storeName();
```

❖ Ví dụ: Gọi Procedure tên là GetAllProducts

```
CALL GetAllProducts();
```

❖ Xem danh sách Stored Procedure trong hệ thống

```
show procedure status;
```

	Db	Name	Type	Definer	Modified	Created	Security_type
	sys	ps_setup_show_enabled	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	ps_setup_show_enabled_consumers	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	ps_setup_show_enabled_instruments	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	ps_statement_avg_latency_histogram	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	ps_trace_statement_digest	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	ps_trace_thread	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	ps_truncate_all_tables	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	statement_performance_analyzer	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	sys	table_exists	PROCEDURE	mysql.sys@localhost	2021-08-19 17:09:48	2021-08-19 17:09:48	INVOKER
	test	GetAllProducts	PROCEDURE	root@localhost	2021-09-22 17:23:59	2021-09-22 17:23:59	DEFINER

Sửa Stored Procedure

- ❖ Xóa Procedure cũ
- ❖ Tạo mới Procedure mới

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS
```

```
`GetAllProducts` $$
```

```
CREATE PROCEDURE `GetAllProducts` ()
```

```
BEGIN
```

```
    SELECT * FROM products;
```

```
END$$
```

```
DELIMITER ;
```

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
	id	title	content
▶	1	he quan tri csdl	Cach tao va su dung stored Procedure
	2	Tutorials học Stored Procedure	Website Tutorials học Stored Procedure



Sửa Stored Procedure

Sửa Stored Procedure có chỉ định User

```
DELIMITER $$  
  
DROP PROCEDURE IF EXISTS `GetAllProducts`$$  
  
CREATE DEFINER=`root`@`localhost` PROCEDURE  
`GetAllProducts`()  
BEGIN  
    SELECT * FROM products;  
END$$  
  
DELIMITER ;
```



Biến trong MySQL Stored Procedure

- ❖ Khai báo biến trong
- ❖ Gán giá trị cho biến
- ❖ Phạm vi hoạt động của biến

❖ Cú pháp :

```
DECLARE variable_name datatype(size) DEFAULT default_value
```

❖ Giải thích:

- **DECLARE:** là từ khóa tạo biến
- **variable_name** là tên biến
- **datatype(size)** là kiểu dữ liệu và kích thước của biến
- **DEFAULT default_value:** là giá trị mặc định cho biến

❖ Ví dụ:

```
DECLARE product_title VARCHAR(255) DEFAULT 'No Name';
```

Gán giá trị cho biến

❖ Cú pháp:

```
SET variable_name = 'value';
```

❖ Ví dụ:

```
DECLARE age INT(11) DEFAULT 0  
SET age = 12
```

```
DECLARE total_products INT DEFAULT 0  
  
SELECT COUNT(*) INTO total_products  
FROM products
```

Phạm vi hoạt động của biến

- ❖ Biến cục bộ của Procedure: biến nằm bên trong phần thân của Procedure .
- ❖ Có thể định nghĩa nhiều biến trong một Procedure.

```
DELIMITER $$  
CREATE PROCEDURE Sum()  
BEGIN  
    DECLARE a INT (11) DEFAULT 0;  
    DECLARE b INT (11) DEFAULT 0;  
    DECLARE tong INT (11) DEFAULT 0;  
  
    SET a = 200;  
    SET b = 300;  
    SET tong = a + b;  
  
    SELECT tong;  
  
END; $$  
DELIMITER;
```

Phạm vi hoạt động của biến

```
DELIMITER $$  
DECLARE tong INT (11) DEFAULT 0;  
DROP PROCEDURE IF EXISTS Sum $$  
CREATE PROCEDURE Sum()  
BEGIN  
    DECLARE a INT (11) DEFAULT 0;  
    DECLARE b INT (11) DEFAULT 0;  
  
    SET a = 200;  
    SET b = 300;  
    SET tong = a + b;  
  
    SELECT tong;  
  
END; $$  
DELIMITER;
```

➔ **Lỗi**



Truyền tham số vào MySQL Stored Procedure

- ❖ Các loại tham số trong MySQL Stored Procedure:
 - **IN**: chế độ mặc định. Khi sử dụng mức này thì nó sẽ được bảo vệ an toàn.
 - **OUT**: Sẽ thay đổi nếu trong Procedure có tác động thay đổi.
 - **INOUT**: kết hợp giữa **IN** và **OUT**

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS getById $$  
  
CREATE PROCEDURE getById(  
    IN id INT(11),  
    IN title VARCHAR(255)  
)  
BEGIN  
    /*Code*/  
END; $$  
DELIMITER;
```



Tham số loại IN trong Mysql Stored Procedure

- ❖ Tham số **INT** sẽ được bảo vệ và không bị thay đổi trong quá trình sử dụng.
- ❖ **Ví dụ:** Viết Store lấy chi tiết sản phẩm theo ID

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS getById $$  
CREATE PROCEDURE getById(IN idVal INT(11))  
BEGIN  
    SELECT * FROM products WHERE id = idVal;  
END; $$  
DELIMITER ;
```

```
CALL getById(1);
```

Result Grid Filter Rows: Export: Wrap Cell Content:			
	id	title	content
▶	1	he quan tri csdl	Cach tao va su dung stored Procedure



Tham số loại OUT trong Mysql Stored Procedure

- ❖ Nếu trong quá trình thực thi mà Procedure có tác động đến tham số này thì bên ngoài nó ảnh hưởng theo.
- ❖ Khi nhận tham số này thì Procedure sẽ hiểu đó là giá trị NULL.
- ❖ Biến truyền vào phải có chữ @ đằng trước
- ❖ **Ví dụ:** Truyền tham số title kiểu **OUT** vào Procedure và đổi giá trị cho nó, sau đó bên ngoài Procedure hiển thị giá trị của title.

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS changeTitle $$  
CREATE PROCEDURE changeTitle(OUT title VARCHAR(255))  
BEGIN  
    SET title = 'Lop CTK42 dang hoc he quan tri csdl';  
END; $$  
DELIMITER ;
```

```
CALL changeTitle(@title);  
SELECT @title;
```

Result Grid		Filter Rows:
	@title	
▶	Lop CTK42 dang hoc he quan tri csdl	



Tham số loại OUT trong Mysql Stored Procedure

- ❖ Khi truyền tham số dạng **OUT** là để lấy dữ liệu trong Procedure và sử dụng ở bên ngoài.
- ❖ Khi truyền tham số vào dạng **OUT** phải có chữ @ đằng trước biến
- ❖ Hoạt động giống tham chiếu nên biến truyền vào dạng **OUT** không cần định nghĩa trước, vì vậy khởi đầu nó có giá trị NULL



Tham số dạng INOUT trong Mysql Stored Procedure

- ❖ Có thể được định nghĩa trước và gán giá trị trước rồi truyền vào Procedure.
- ❖ Sau khi thực thi xong nếu trong Procedure có tác động đến thì ảnh hưởng theo.

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS counter $$  
  
CREATE PROCEDURE counter(INOUT number INT(11))  
BEGIN  
    SET number = number + 1;  
END; $$  
DELIMITER;
```

```
SET @counter = 1;  
CALL counter(@counter);  
SELECT @counter;
```

→ 2



Câu lệnh IF ELSE trong MYSQL

- ❖ Cho phép tạo luồng xử lý rẽ nhánh, nếu đúng thì thực thi và ngược lại mệnh đề sai thì sẽ không thực thi.
- ❖ Thông thường có thể kết hợp các toán tử, toán hạng và biến trong MYSQL để tạo ra các mệnh đề đúng sai trong điều kiện của lệnh IF.
- ❖ **Cú pháp:**

```
IF if_expression THEN  
    commands  
ELSEIF elseif_expression THEN  
    commands  
ELSE  
    commands  
END IF;
```

Câu lệnh IF ELSE trong MYSQL

❖ Ví dụ:

```
CREATE TABLE IF NOT EXISTS `members` (  
  `us_id` INT(11) NOT NULL AUTO_INCREMENT,  
  `us_username` VARCHAR(30) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `us_password` VARCHAR(32) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `us_level` TINYINT(1) DEFAULT '0',  
  PRIMARY KEY (`us_id`)  
) ENGINE=INNODB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=4 ;
```

```
INSERT INTO `members` (`us_id`, `us_username`, `us_password`,  
  `us_level`) VALUES  
(1, 'admin', '0ef218d727d08b85cf24e262ff37a23e', 1),  
(2, 'member', '0ef218d727d08b85cf24e262ff37a23e', 2),  
(3, 'banded', '0ef218d727d08b85cf24e262ff37a23e', 0);
```



Câu lệnh IF ELSE trong MYSQL

❖ Viết Procedure đăng nhập với yêu cầu như sau:

- Nếu `us_level = 0` => tài khoản bị khóa
- Nếu `us_level = 1` => là admin
- Nếu `us_level = 2` => là member
- Nếu không tồn tại => đăng nhập sai



Câu lệnh IF ELSE trong MYSQL

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS `checkLogin`$$
```

```
CREATE PROCEDURE `checkLogin`(  
    IN input_username VARCHAR(255),  
    IN input_password VARCHAR(255),  
    OUT result VARCHAR(255)  
)  
BEGIN
```

```
    /*Bien flag lưu trữ level. Mặc định là -1*/  
    DECLARE flag INT(11) DEFAULT -1;
```

```
    /*Thực hiện truy vấn gán level vào biến flag*/  
    SELECT us_level INTO flag FROM members  
    WHERE us_username = input_username AND us_password = MD5(input_password);
```

```
    /*Sau khi thực hiện lệnh select mà không có dữ liệu thì lúc này flag sẽ không thay đổi*/  
    IF (flag <= 0) THEN
```

```
        SET result = 'Thông tin đăng nhập sai';
```

```
    ELSEIF (flag = 0) THEN
```

```
        SET result = 'Tai khoản bị khóa';
```

```
    ELSEIF (flag = 1) THEN
```

```
        SET result = 'Tai khoản admin';
```

```
    ELSE
```

```
        SET result = 'Tai khoản member';
```

```
    END IF;
```

```
END$$
```

```
DELIMITER ;
```



Câu lệnh IF ELSE trong MYSQL

```
CALL checkLogin('admin', 'ctk42', @result);  
SELECT @result;
```

```
CALL checkLogin('member', 'ctk42', @result);  
SELECT @result;
```

```
CALL checkLogin('banded', 'ctk42', @result);  
SELECT @result;
```

❖ Cú pháp:

```
CASE case_expression
  WHEN when_expression_1 THEN commands
  WHEN when_expression_2 THEN commands
  ...
  ELSE commands
END CASE;
```

- ❖ **Ví dụ:** Viết Stored Procedure nhập vào một số từ 0 tới 9 và đọc số đó thành chữ. Yêu cầu sử dụng lệnh CASE trong MYSQL



Câu lệnh CASE – Ví dụ

```
DELIMITER $$
DROP PROCEDURE IF EXISTS `docSo`$$
CREATE PROCEDURE `docSo`(IN so INT(11))
BEGIN
    DECLARE message VARCHAR(255);
    CASE so
        WHEN 0 THEN
            SET message = 'KHONG';
        WHEN 1 THEN
            SET message = 'MOT';
        WHEN 2 THEN
            SET message = 'HAI';
        WHEN 3 THEN
            SET message = 'BA';
        WHEN 4 THEN
            SET message = 'BON';
        WHEN 5 THEN
            SET message = 'NAM';
        WHEN 6 THEN
            SET message = 'SAU';
        WHEN 7 THEN
            SET message = 'BAY';
        WHEN 8 THEN
            SET message = 'TAM';
        WHEN 9 THEN
            SET message = 'CHIN';
        ELSE
            SET message = 'KHONG TIM THAY';
    END CASE;
    SELECT message;
END$$
DELIMITER ;
```

Câu lệnh CASE – Ví dụ

```
CALL docSo(1); -- MOT
CALL docSo(2); -- HAI
CALL docSo(3); -- BA
CALL docSo(4); -- BON
CALL docSo(5); -- NAM
CALL docSo(6); -- SAU
CALL docSo(7); -- BAY
CALL docSo(8); -- TAM
CALL docSo(9); -- CHIN
CALL docSo('muoi'); -- KHONG TIM THAY
```



Vòng lặp while trong MySQL

❖ Cú pháp:

```
WHILE expression DO  
    Statements  
END WHILE;
```

- ❖ **Ví dụ:**Viết chương trình Stored Procedure in ra màn hình các số từ a tới b bằng vòng lặp While.

Vòng lặp while trong MySQL

❖ **Ví dụ:**Viết chương trình Stored Procedure in ra màn hình các số từ a tới b bằng vòng lặp While.

```
DELIMITER $$
DROP PROCEDURE IF EXISTS loopWhile$$
CREATE PROCEDURE loopWhile(
    IN a INT(11),
    IN b INT(11)
)
BEGIN
    -- Khai bao Chuoi in ra man hinh--
    DECLARE str VARCHAR(255) DEFAULT '';
    WHILE (a <= b) DO
        SET str = CONCAT(str,a,',');
        SET a = a + 1;
    END WHILE;
    SELECT str;
END$$
DELIMITER ;
```

Vòng lặp while trong MySQL

- ❖ **Ví dụ:**Viết chương trình Stored Procedure in ra màn hình các số từ a tới b bằng vòng lặp While.

```
CALL loopWhile(1,10);
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	str		
▶	1,2,3,4,5,6,7,8,9,10,		

