

# 一. 基本概念

1. 什么是存储过程：类似于Pascal中的过程概念，存储过程就是一条或多条MySQL语句的集合。
2. 存储过程的优点：
  - ①简单：把处理封装到容易使用的单元中。
  - ②安全：简化了对于变动的管理。少了数据讹误。
  - ③高性能

# 二. 使用存储过程

## I. 执行存储过程

1. 使用CALL关键字（“调用”）：`CALL procedurename (paras) ;`
2. 关于参数：
  - ①变量：所有变量名之前都要加上@
  - ②常量：常量该怎么写就怎么写

## II. 定义存储过程

### 1. 基本格式：

```
CREATE procedurename(  
    IN para1 type,  
    OUT para2 type  
)  
BEGIN  
    SELECT ...  
    INTO para2;  
END//
```

### 2. 详细解释：

- ①参数：其实一目了然，IN代表要输入的量，而OUT代表要输出的量，注意在最后给出相关的类型
- ②函数体：用BEGIN和END；
- ③将结果存到OUT的参数中：使用INTO关键字。

**3. 注意：**如果（比如命令行）使用程序要解释存储过程自身的；字符。那么我们在过程体内输入；的时候就会报错。这一个BUG的解决方法是临时更改命令行实用程序的**语句分隔符**：

```
DELIMITER //
```

(注意这一改就会全部生效了，所以在写完PROCEDURE后还要再改回来:DELIMITER ;)

**4.使用输出的变量：**要注意这里输出的变量并不是临时的。我们可以用SELECT来调用里面的值

```
SELECT @aa;
```

这个语句就可以显示之前aa存储的计算结果了。

### III. 删除存储过程

```
DROP PROCEDURE procedurename (IF EXISTS) ;
```

## 三. 使用更加高级的存储过程

这个更加高级体现在两个方面：

1. 使用DECLEAR声明了局部变量
2. 使用IF-THEN-ELSE-ENDIF等语句进行逻辑判断，当然同时引入了布尔型变量

使用示例：

```
mysql> DELIMITER //
```

```
mysql> CREATE PROCEDURE ordertotal(
```

```
->     IN number INT,
```

```
->     IN texable BOOLEAN,
```

```
->     OUT ototal DECIMAL(8, 2)
```

```
-> ) COMMENT 'Obtain ordertotal, optionally adding tax'
```

```
-> BEGIN
```

```
-> -- Declare variables for total
```

```
-> DECLARE total DECIMAL(8, 2);
```

```
-> -- Remember how to spell 'Declare'
```

```
-> DECLARE taxrate INT DEFAULT 6;
```

```
->
```

```
-> -- Get the order total
```

```
-> SELECT SUM(item_price * quantity)
```

```
-> FROM orderitems
```

```
-> WHERE order_num = number
```

```
-> INTO total;
```

```
->
```

```
-> -- Is this taxable?
```

```
-> IF texable THEN
```

```
->     SELECT total + (total * taxrate / 100) INTO total;
```

```
-> END IF;  
->  
-> SELECT total INTO ototal;  
-> END//
```

讲解一下使用技巧：

1. **多用注释**：在mysql中，注释很简单: --; 最好在注释后面添加必要的空格；
2. 在参数列表和BEGIN 之间**使用COMMENT关键字**，后面写一个字符串（别忘了 “ ! ） ， 用来表示该过程的作用
3. 注意DECLARE的写法。。防止写错
4. 注意在使用IF的时候，后面的条件不需要使用()括起来

最后显示结果：

```
CALL order total (20005, 0, @total) ;
```

```
SELECT @total;
```

（注意在参数列表中还要给出存储结果的参数）

## 四. 查看存储过程

1. 存储过程实在是太长了，有时我们都忘了存储过程中的细节了。这时我们可以使用SHOW CREATE PROCEDURE *procedurename*; 来查看原来的语句。
2. 查看存储过程列表：我们可能会创建多个存储过程，mysql为我们准备了一个存储多个过程的列表，我们可以使用SHOW PROCEDURE STATUS;来查看。