每日一练6 存储过程.md

编程要求

根据提示，在右侧编辑器补充代码，创建存储过程`GetCustomerLevel(in p_customNumber int(11),out p_customerLevel varchar(10))`，通过查询`customers`表中客户的信用额度，来决定客户级别，并将客户编号和对应等级输出，具体输出内容参考测试集。

`customers`表数据结构：

customerNumber	creditlimit
101	2000
102	12000
103	6000

等级设定：

```
PLATINUM （creditlim>10000）
GOLD （5000<=creditlim<=10000）
SILVER（creditlim<5000）
```

提示：你可能需要使用到定义变量和判断，

1. 变量的定义和使用：

```
declare 变量名 类型; #定义变量
select id into 变量名 from table; #将table表中的id列值赋给变量。
```

2. `SQL`中的`if`和`case`语法：

```
IF expression THEN
 statements;
ELSEIF elseif-expression THEN
 elseif-statements;
...
ELSE
 else-statements;
END IF;;
```

```
CASE case-expression
 WHEN when_expression_1 THEN commands
 WHEN when_expression_2 THEN commands
 ...
 ELSE commands
END CASE;
```

答案

```
USE mydb;
#请在此处添加实现代码
########## Begin ##########
DELIMITER //
CREATE PROCEDURE GetCustomerLevel(in p_customNumber int(11), out p_customerLevel
varchar(10))
Begin
    DECLARE creditlim double;
    select creditlimit into creditlim from customers where customerNumber =
p_customNumber;
    IF creditlim > 10000 THEN
    SET p_customerLevel = 'PLATINUM';

    ELSEIF creditlim >= 5000 and creditlim <= 10000 THEN
    SET p_customerLevel = 'GOLD';

    ELSE SET p_customerLevel =  'SILVER';
    END IF;
    SELECT  p_customNumber as customerNumber, p_customerLevel;

END //
DELIMITER ;

########## End ##########
```

解析

存储过程的定义

存储过程（`Stored Procedure`）是一种在数据库中存储复杂程序，以便外部程序调用的一种数据库对象。

存储过程是为了完成特定功能的 `SQL` 语句集，经编译创建并保存在数据库中，用户可通过指定存储过程的名字并给定参数（需要时）来调用执行。

存储过程思想上很简单，就是数据库 `SQL` 语言层面的代码**封装与重用。**

简单的说存储过程就是具有名字的一段代码，用来完成一个特定的功能。

存储过程的创建和查询

创建存储过程：`create procedure 存储过程名(参数)`

- 下面我们来创建第一个存储过程

每个存储的程序都包含一个由 `SQL` 语句组成的主体。此语句可能是由以分号（`;`）字符分隔的多个语句组成的复合语句。例如：

```
CREATE PROCEDURE proc1()
BEGIN
SELECT * FROM user;
END;
```

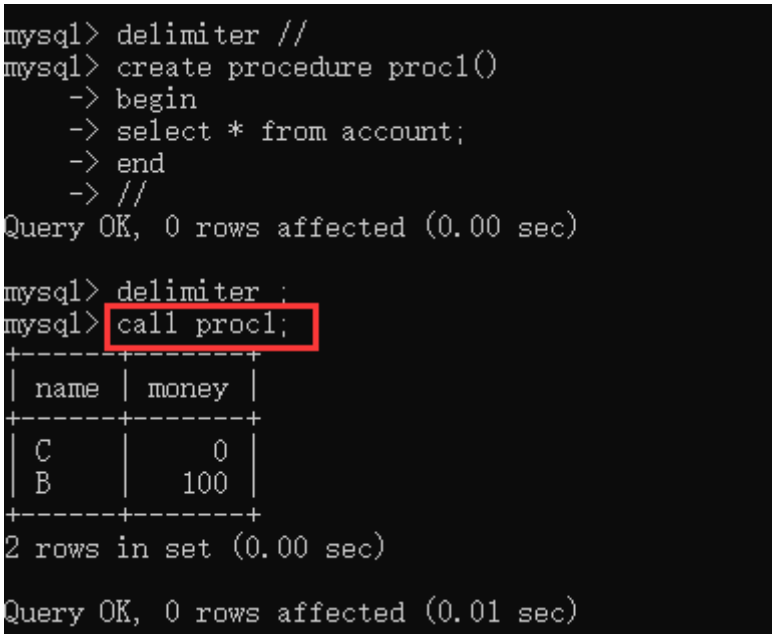
`MySQL` 本身将分号识别为语句分隔符，因此必须临时重新定义分隔符以使 MySQL 将整个存储的程序定义传递给服务器。

要重新定义 `MySQL` 分隔符，请使用该 `delimiter` 命令。使用 `delimiter` 首先将结束符定义为 `//`，完成创建存储过程后，使用 `//` 表示结束，然后将分隔符重新设置为分号（`;`）：

```
DELIMITER //
CREATE PROCEDURE proc1()
BEGIN
SELECT * FROM user;
END //
DELIMITER ;
```

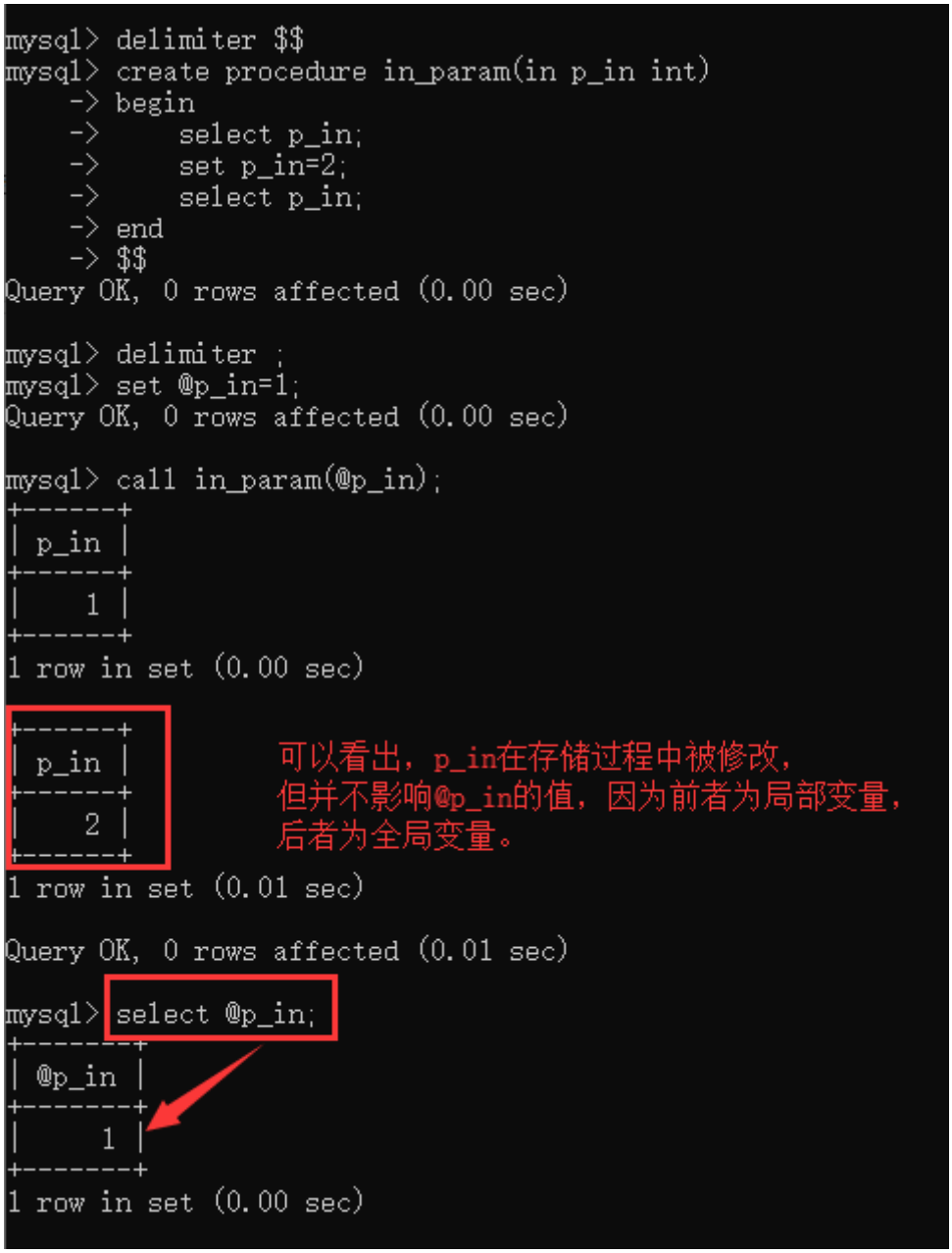
注意：`/` 也可以换成其他符号，例如 `$`；

- 执行存储过程：`call 存储过程名`

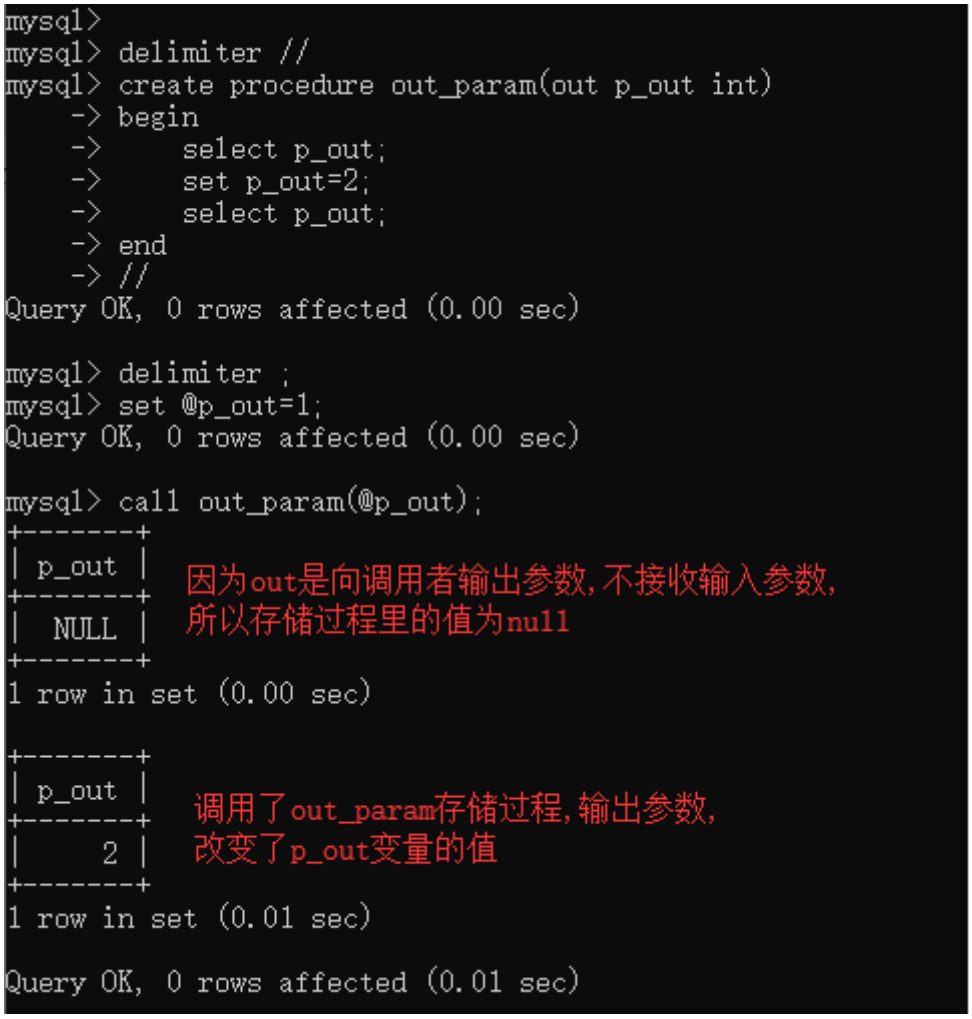


- 创建带有参数的存储过程 存储过程的参数有三种：
  - `IN`：输入参数，也是默认模式，表示该参数的值必须在调用存储过程时指定，在存储过程中修改该参数的值不能被返回；
  - `OUT`：输出参数，该值可在存储过程内部被改变，并可返回；
  - `INOUT`：输入输出参数，调用时指定，并且可被改变和返回。

`IN` 参数示例：


mysql> delimiter $$
mysql> create procedure in_param(in p_in int)
-> begin
-> select p_in;
-> set p_in=2;
-> select p_in;
-> end
-> $$
Query OK, 0 rows affected (0.00 sec)
mysql> delimiter ;
mysql> set @p_in=1;
Query OK, 0 rows affected (0.00 sec)
mysql> call in_param(@p_in);
| p_in |
| 1 |
1 row in set (0.00 sec)
| p_in |
| 2 |
1 row in set (0.01 sec)
Query OK, 0 rows affected (0.01 sec)
mysql> select @p_in;
| @p_in |
| 1 |
1 row in set (0.00 sec)
可以看出，p_in在存储过程中被修改，
但并不影响@p_in的值，因为前者为局部变量，
后者为全局变量。

`OUT` 参数示例：



存储过程的查询和删除

我们如何在数据库中查询我们已经创建过的存储过程呢：

```
SHOW PROCEDURE STATUS WHERE db='数据库名';
```

查看存储过程的详细定义信息：

```
SHOW CREATE PROCEDURE 数据库.存储过程名;
```

当我们不再需要某个存储过程时，我们可以使用：

```
DROP PROCEDURE [IF EXISTS] 数据库名.存储过程名;
```