
创建存储过程

可以使用 CREATE PROCEDURE 语句创建存储过程，语法格式如下：

```
CREATE PROCEDURE <过程名> ( [过程参数[, ...] ] ) <过程体>
```

[过程参数[, ...]] 格式如下：

```
[ IN | OUT | INOUT ] <参数名> <类型>
```

语法说明如下：

1. 过程名

存储过程的名称，默认在当前数据库中创建。若需要在特定数据库中创建存储过程，则要在名称前面加上数据库的名称，如 db_name.pr_name。

注意：名称应当尽量避免选取与 MySQL 内置函数相同的名称，否则会发生错误。

2. 过程参数

存储过程的参数列表。其中，<参数名>为参数名，<类型>为参数的类型（可以是任何有效的 MySQL 数据类型）。当有多个参数时，参数列表中彼此间用逗号分隔。存储过程可以没有参数（此时存储过程的名称后仍需加上一对括号），也可以有 1 个或多个参数。

MySQL 存储过程支持三种类型的参数，即输入参数、输出参数和输入/输出参数，分别用 IN、OUT 和 INOUT 三个关键字标识。其中，输入参数可以传递给一个存储过程，输出参数用于存储过程需要返回一个操作结果的情形，而输入/输出参数既可以充当输入参数也可以充当输出参数。

注意：参数的取名不要与数据表的列名相同，否则尽管不会返回出错信息，但是存储过程的 SQL 语句会将参数名看作列名，从而引发不可预知的结果。

3. 过程体

存储过程的主体部分，也称为存储过程体，包含在过程调用的时候必须执行的 SQL 语句。这个部分以关键字 BEGIN 开始，以关键字 END 结束。若存储过程体中只有一条 SQL 语句，则可以省略 BEGIN-END 标志。

在 MySQL 中，服务器处理 SQL 语句默认是以分号作为语句结束标志的。然而，在创建存储过程时，存储过程体可能包含有多条 SQL 语句，这些 SQL 语句如果仍以分号作为语句结束符，那么 MySQL 服务器在处理时会以遇到的第一条 SQL 语句结尾处的分号作为整个程序的结束符，而不再去处理存储过程体中后面的 SQL 语句，这样显然不行。

所以，在存储过程中需要使用 DELIMITER 命令将结束命令修改为其他字符。语法格式如下：

```
DELIMITER $$
```

语法说明如下：

\$\$ 是用户定义的结束符，这个符号也可以是其他特殊符号，如两个“?”或两个“¥”等。要避免使用反斜杠“\”字符，因为它是 MySQL 的转义字符。

当在 MySQL 命令行客户端输入如下 SQL 语句时：

```
mysql > DELIMITER $$
```

成功执行这条 SQL 语句后，任何命令、语句的结束标志就换为“\$\$”了。

若希望换回默认的分号“;”作为结束标志，则在 MySQL 命令行客户端输入下列语句：

```
mysql > DELIMITER ;
```

注意：DELIMITER 和分号“\$\$”或者“;”之间一定要有一个空格。

【例 8-13】编写一个存储过程，按照提供的读者姓名，查询出该读者的相关信息。代码如下：

```
delimiter $$
```

```
create procedure pr_showReader(in v_name varchar(10))
begin
    select * from reader where readerName=v_name;
end$$
delimiter ;
```

```
mysql> delimiter $$
mysql> create procedure pr_showReader(in v_name varchar(10))
    -> begin
    -> select * from reader where readerName=v_name;
    -> end $$
Query OK, 0 rows affected (0.17 sec)
mysql> delimiter ;
```

结果显示 pr_showReader 存储过程已经创建成功。