## 存储过程的概念

类似于一个函数，将常用的sql语句封装起来，形成一个sql语句集。在第一次调用后进行编译，之后再调用就不用编译，可以直接使用，效率比较高。

将重复执行的sql语句封装成一个集合，只编译一次，后面再次需要就直接调用，不需要再编译。

## 存储过程的创建

Create procedure 存储过程名称 （[ in | out | inout ] 参数名称 参数类型）

Begin

过程体

End

In: 表示输入参数，即参数是我们提供的

Out: 表示输出参数，类似于函数的返回值，初始值为null

Inout：在调用前，将参数传入，完成后，将结果放入

## 存储过程的察看和调用

Show create procedure 存储过程名称

Show procedure status [ like匹配模式 ]

调用：call 数据库名.存储过程名称 （ 参数列表 ）

如果参数是in, 实参为变量或者直接的数据

如果参数是out/inout，参数要是一个变量

## 存储过程的修改和删除

修改：alter procedure 存储过程名称 [ 特征 ]

特征是指修改那个部分。这种修改不能修改参数和主体

Comment ‘修改注释’

Language sql 修改使用的语句

删除：

Drop procedure 存储过程名称

## 变量

**系统变量**，对所有客户端有限

察看：show global variables [ like匹配模式 | where 表达式 ]

修改：局部修改，修改只在本次连接中有效；全局修改，对已经连接的客户端无效，但是对新连接的客户端永久有效。

局部修改：set 变量名 = 新值

全局修改：方法一 set global 变量名 = 新值； 方式二 set @@global.变量名 = 新值；

**会话变量: 只有会话变量有@变量名组成。会话变量才加@**

会话变量也称为用户变量，是用户自定义的变量，系统变量为系统定义的变量。会话变量是和当前客户端绑定的，仅仅对当前客户端有效。

会话变量的名称：@符号+变量名：eg: @name

赋值：利用set语句，利用赋值符号select配合 ：=， 利用select into 语句。必须为会话变量赋初始值

Set @name = ‘zhangkuan’

Select @price : = id from t limit 3 (将最后一条记录的值附上)

Select id from t limit 1

Into @price (必须只能是一条记录被查出来)

**局部变量：**

只在begin 和 end之间有效，在其他地方无效

具体语法格式：

Declare 变量名1 数据类型 [ default 默认值 ]

Declare age int default 10;