- 5.7. MySQL中的函数与过程(补充)
- 5.7.1. 基本概念

函数与过程合称为存储过程SP(Stored Procedure), SP可以理解为具备了SQL功能的结构化程序设计语言, 包含了顺序、分支、循环等逻辑控制结构,并将SQL语 句封装在SP内部,由MySQL服务器直接存储和执行,具 有提高运行速度、提高数据的安全性、减少代码冗余等 特点

- ★ 可被应用程序直接调用
- ★ SP在不同关系型数据库之间的移植比较困难 (不属于标准SQL, 各DBMS的定义方式均不相同)
- ★ 触发器也可以理解为SP(自动执行,不能被调用)

- 5.7. MySQL中的函数与过程(补充)
- 5.7.2. 函数的使用
- 5.7.2.1. 函数的定义

create function 函数名(形参表) returns 返回值类型

begin

函数体

- ★ 函数调用可出现在所有的SQL命令中
- ★ 输入参数只能是值参数形式(类似于C++的单向传值)
- ★ 函数只能返回一个值(返回类型用returns指定)
- ★ 函数体中不能是运行结果为集合的语句(select)

- 5.7. MySQL中的函数与过程(补充)
- 5.7.2. 函数的使用
- 5.7.2.1. 函数的定义

create function 函数名(形参表) returns 返回值类型

begin

函数体

- ★ 函数可以嵌套调用(自定义函数/系统函数均可)
- ★ 定义函数时需要重新设置语句结束定义符

- 5.7. MySQL中的函数与过程(补充)
- 5.7.2. 函数的使用
- 5.7.2.1. 函数的定义

create function 函数名(形参表) returns 返回值类型

begin

函数体

end

- 5.7.2.2.显示系统中已定义的函数 show function status
- 5.7.2.3. 删除已定义的函数 drop function 函数名

★ 不提供修改功能,
通过删除后重建的方式进行



- 5.7. MySQL中的函数与过程(补充)
- 5.7.3. 过程的使用
- 5.7.3.1. 过程的定义

create procedure 过程名(形参表)

begin

过程体

- ★ 过程调用通过 "call 过程名(实参表)" 的形式
- ★ 形参表可以是值参数形式(类似于C++的单向传值), 也可以是引用参数形式(可返回值的参数,类似于C++ 中的指针参数),且均允许0-n个,通过IN(值参数)/ OUT(引用参数)/INOUT(值+引用)来指定参数的类型

- 5.7. MySQL中的函数与过程(补充)
- 5.7.3. 过程的使用
- 5.7.3.1. 过程的定义

create procedure 过程名(形参表)

begin

过程体

- ★ 过程体中的语句可以返回一个或多个select的结果
- ★ 过程可以嵌套调用其它过程及函数
- ★ 定义过程时需要重新设置语句结束定义符

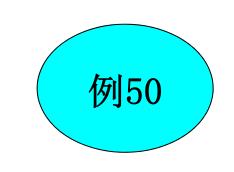
- 5.7. MySQL中的函数与过程(补充)
- 5.7.3. 过程的使用
- 5.7.3.1. 过程的定义

create procedure 过程名(形参表)

begin

过程体

- 5.7.3.2. 显示系统中已定义的过程 show procedure status
- 5.7.3.3.删除已定义的过程 drop procedure 过程名
- ★ 不提供修改功能,通过删除后重建的方式进行



- 5.7. MySQL中的函数与过程(补充)
- 5.7.4. 函数与过程比较

	函数	过程
调用方式	嵌在所有SQL命令中	只能通过call命令
返回值	只能返回一个值	可返回多个select 结果
参数	只允许值参数	值参数和引用参数 均可使用
定义中可用语句	不允许返回集合的 SQL命令	几乎所有SQL命令
调用其它函数及过 程	只允许调用函数,不 允许调用过程	可调用其它函数及 过程