**mysql存储过程**

存储过程概述

存储过程介绍

指的是数据库中保存的一系列sql命令的集合

可以在存储过程中使用变量，条件判断，流程控制等

存储过程，相当于是mysql中的脚本

存储过程优点

提高性能

可减轻网络负担

可以防止对表的直接访问

避免重复编写sql操作

基本使用

创建存储过程

语法格式

mysql>delimiter//

mysql>create procedure 名称()

->begin

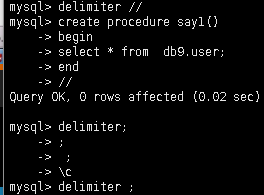
->....功能代码

->end

->// //结束存储过程

mysql>-delimiter;

delimiter关键字用来指定存储过程的分隔符（默认为;）



若没有指定分隔符，编译器会把存储过程当成SQL语句进行处理，从而执行出错

查看存储过程

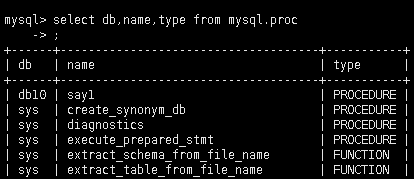
方法1

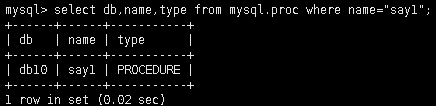
mysql>show procedure status;

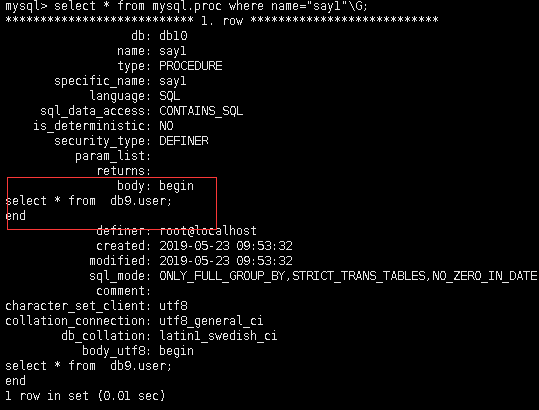


方法2

mysql>select db,name,type from mysql.proc where name=”存储过程名”;







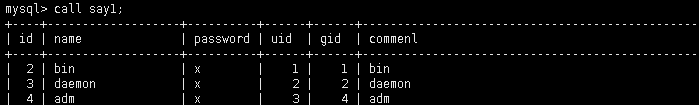
调用/删除存储过程

调用存储过程

call 存储过程名();

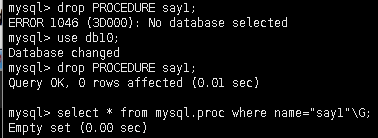
存储过程没有参数时，()可以省略

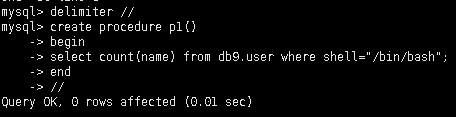
存储过程由参数时，调用时必须传给参数



删除存储过程

drop procedure 存储过程名;





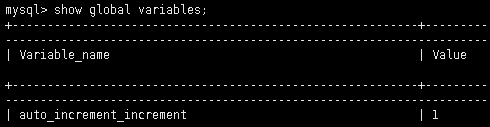
存储过程进阶

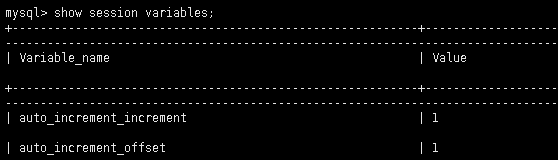
变量类型

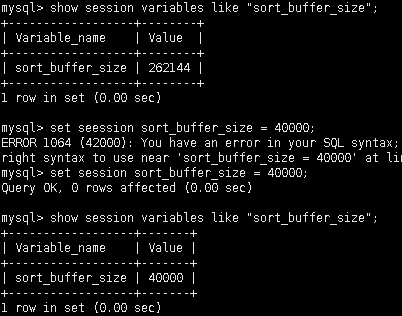
调用局部变量时，变量名前不需要加@

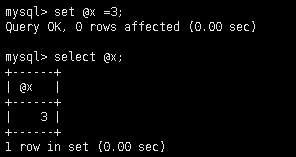
|  |  |
| --- | --- |
| 名称 | 描述 |
| 会话变量 | 会话变量和全局变量叫系统变量 使用set命令定义全局变量的修改会影响到整个服务器，但是对会话变量的修改，只会影响当前的会话  select @@hostname |
| 全局变量 |
| 用户变量 | 在客户端连接到数据库服务的整个过程中都是有效的。当当前连接断开后所有用户变量失效  定义set @变量名=值;  输出 select @变量名; |
| 局部变量 | 存储过程中的begin/end。其有效范围仅限于该语句块中，语句执行完毕后，变量失效  declare专门用来定义局部变量 |

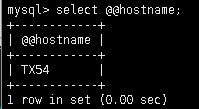


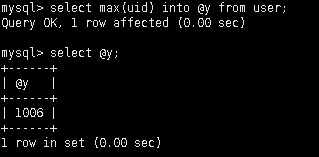


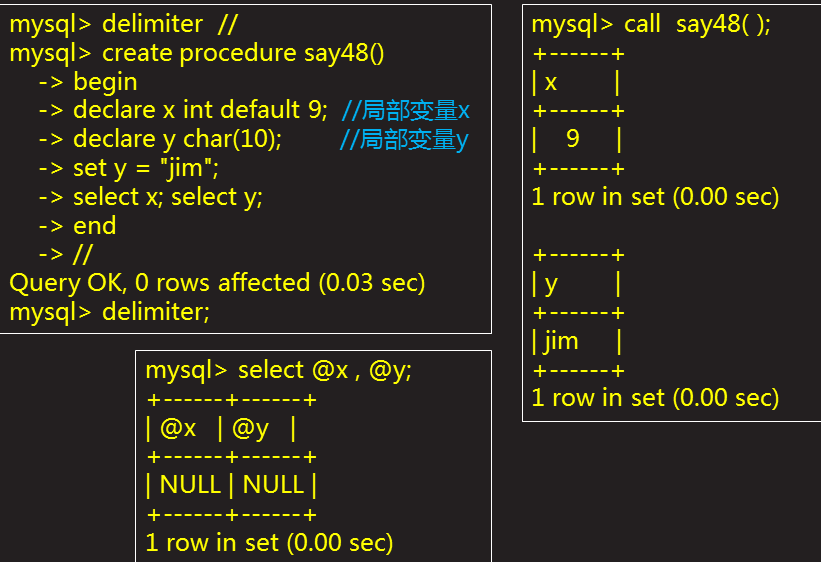


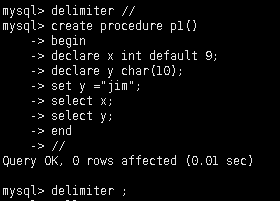


查看用户自定义变量

查看系统变量

使用sql命令结果赋值





参数类型

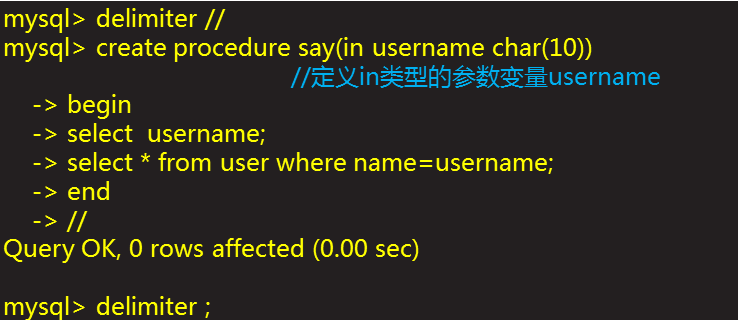
调用参数时，名称前也不需要加@

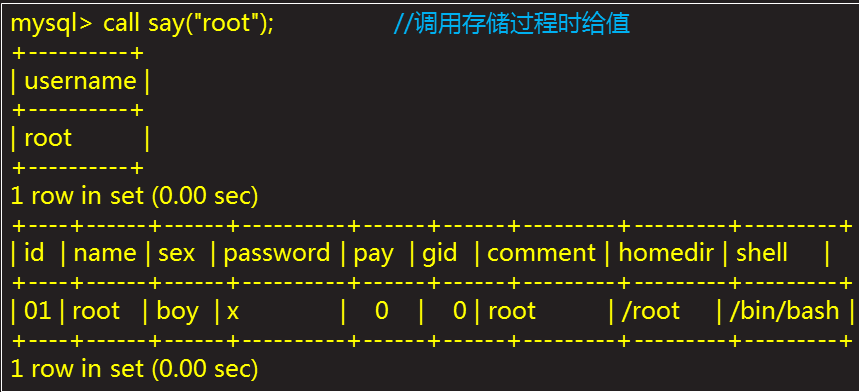
create procedure 名称(

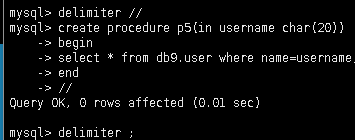
类型 参数名 数据类型,类型 参数名 数据类型

)

|  |  |  |
| --- | --- | --- |
| 关键字 | 名称 | 描述 |
| in | 输入参数 | 作用是给存储过程传值，必须在调用存储过程时赋值，在存储过程中该参数的值不允许修改;默认类型是in |
| out | 输出参数 | 该值可在存储过程内部改变，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 |





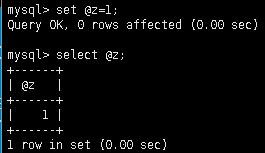


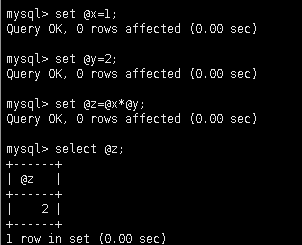


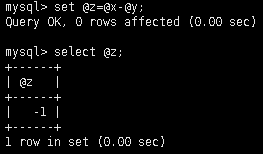
算数运算

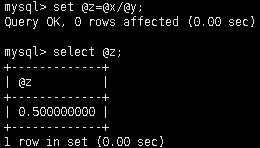
算数符号及用法示例

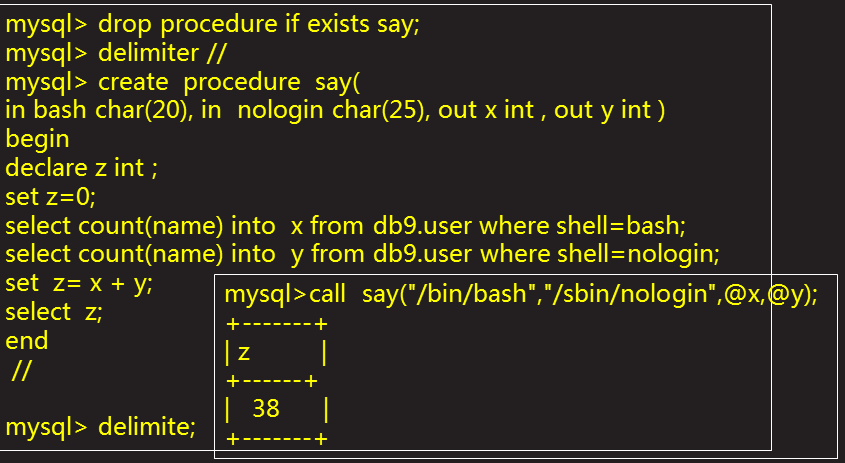
|  |  |  |
| --- | --- | --- |
| 符号 | 描述 | 示例 |
| + | 加法运算 | set @var1=2+2;select @var1; |
| - | 减法运算 | set @var2=3-2;select @var2; |
| \* | 乘法运算 | set @var3=3\*2;select @var3; |
| / | 除法运算 | set @var4=3/2;select @var4; |
| DIV | 整除运算 | set @var5=10 DIV 3;select @var5; |
| % | 取模 | set @var6=10%3;select @var6; |

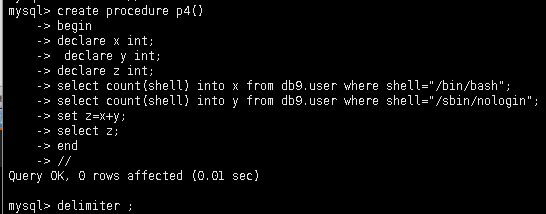


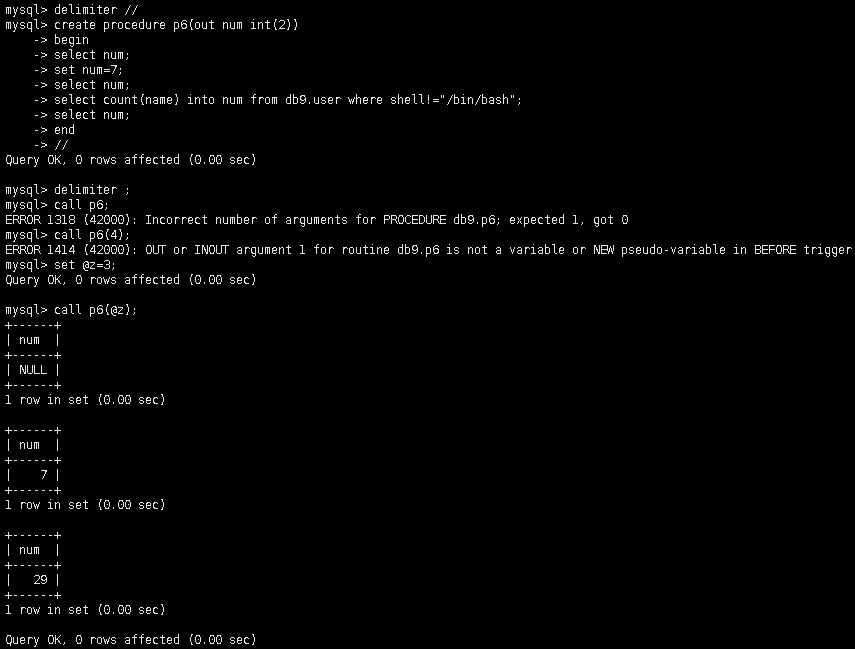


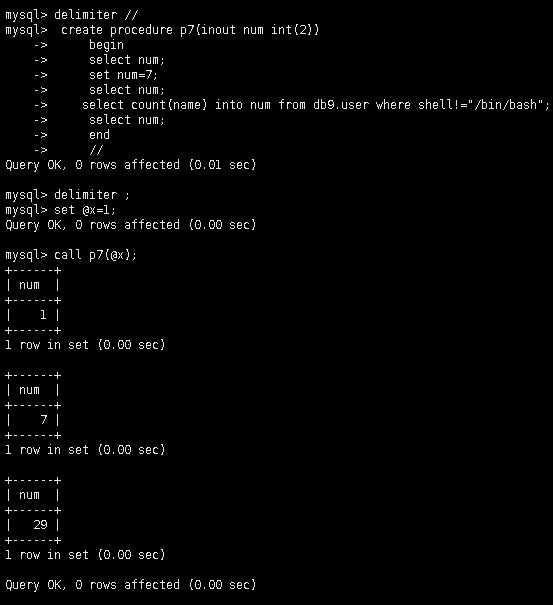












流程控制

条件测试

数值比较

|  |  |
| --- | --- |
| 类型 | 用途 |
| = | 等于 |
| >、>= | 大于、大于或等于 |
| <、<= | 小于、小于或等于 |
| != | 不等于 |
| between...and... | 在..与..之间 |

逻辑比较、范围、空、非空、模糊、正则

|  |  |
| --- | --- |
| 类型 | 用途 |
| OR AND ! | 与 或 非 |
| IN.. NOT IN.. | 在..范围内 不再..范围内 |
| IS NULL | 字段的值为空 |
| IS NOT NULL | 字段的值不为空 |
| like | 模糊匹配 |
| regexp | 正则匹配 |

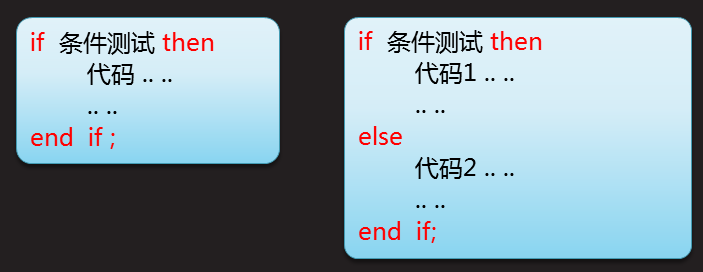
选择结构

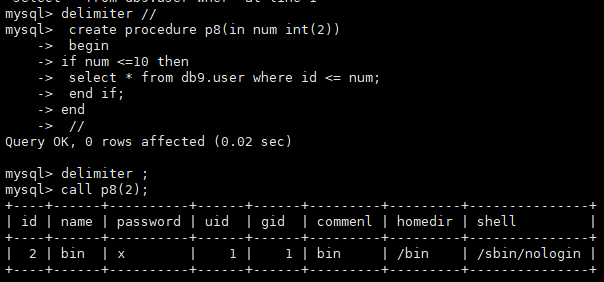
单分支选择结构

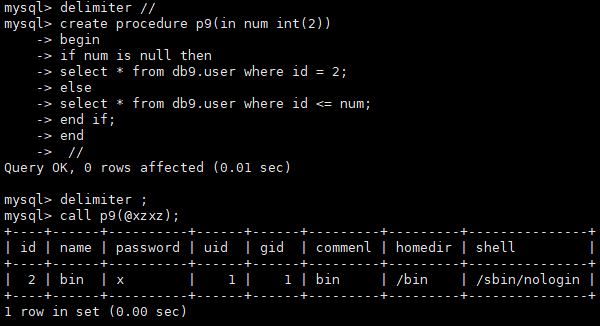
当“条件成立”时执行命令

双分支选择结构

当“条件成立”时执行代码1;否则执行代码2



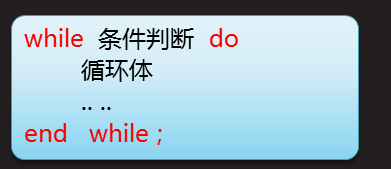


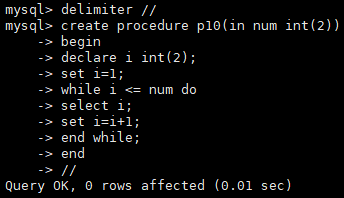


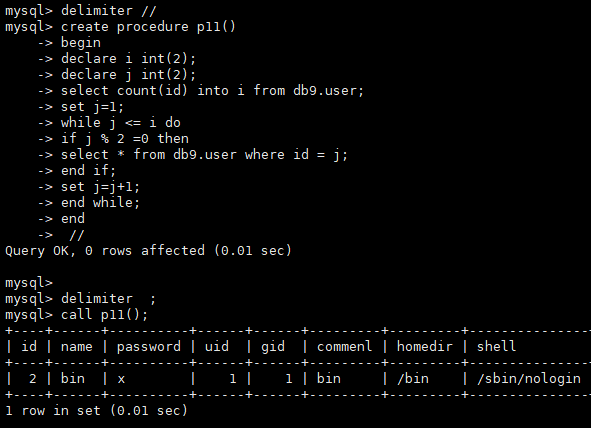
循环结构

while条件式循环

反复测试条件，只要成立就执行命令序列

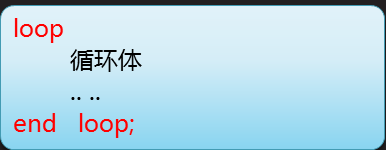


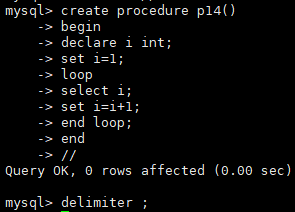


输出为偶数的记录

loop死循环

无条件、反复执行某一段代码

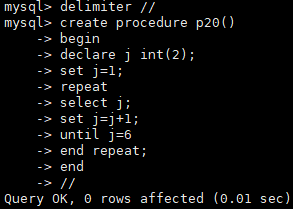




repeat条件式循环

当条件成立时结束循环





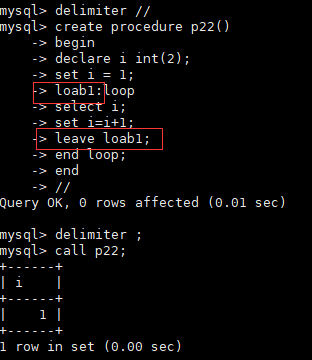


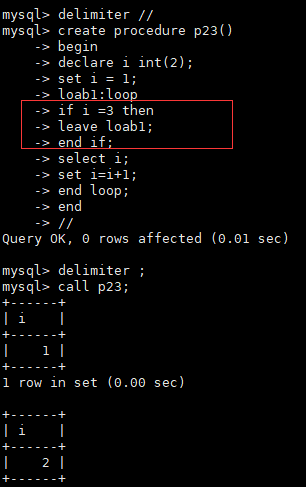
控制循环结构的执行（控制语句）

循环结构控制语句，控制循环结构的执行

leave 标签名 //跳出循环

iterate 标签 //放弃本次循环，并开始下一次循环

跳出所有循环，只执行一次

一般loop与leave连用

