Mysql变量：

1. 系统变量

全局变量 、会话变量

1. 自定义变量

用户变量 、 局部变量

一、系统变量：

注：如果是全局变量，需要加global，如果是会话变量需要加session，不加默认为session。

是由系统提供，不是用户定义的，属于服务器层面。

使用语法：

1. 查看所有的系统变量

show global/【session】variables;

1. 查看满足条件的部分系统变量

show global/【session】variables like ‘%char%’; (例：查看字符的部分系统变量)

1. 查看指定的某个系统变量

Select @@变量名；

1. 设置系统变量

注：（全局变量的作用域：针对所有的会话<链接>有效，但不能夸重启（重启有效需要去设置配置文件）

会话变量的作用域：只是针对当前会话。）

方式一

Set global/【session】变量名=值；

方式二

Set @@ global/【session】.变量名=值；

二、自定义变量

用户自己定义的变量，作用域针对当前的会话（链接）有效，同与会话变量的作用域

1、用户变量

1）声明（不需要设置类型）

①声明并初始化：

Set @用户变量名=值； set @用户变量名:=值； select @用户变量名:=值

2）赋值

方式一：

Set @用户变量名=值； set @用户变量名:=值； select @用户变量名:=值

方拾二：

Select 字段名 into @用户变量名 from 表明；

（将表中字段值赋给变量，该字段只能有一个值）

3）使用（查看，比较，运算等）

查看用户变量：select @用户变量名；

例 ：声明并初始化set @name=’joke’; 查看用户变量 select @name；

Set @name=2; select @name; 结果：2

Set @a=1; set @b=3; set @sum =@a+@b; select @sum=4;

字段赋值：Select count(\*) into @conut from employees;

2、局部变量

作用域：自在定义他得begin end中有效。 应用在begin end中 而且在第一句话。

1. 声明

Declare 变量名 类型；

Declare 变量名 类型 default 值；

1. 赋值

方式一：

Set 局部变量名=值； set 局部变量名:=值； select @局部变量名:=值

方拾二：

Select 字段名 into 局部变量名 from 表明；

1. 使用

查看：select 局部变量名；

三、存储过程和函数

存储过程和函数类似于java中的方法。

1. 存储过程

一组预先编译好的sql语句的集合

1. 创建语法

Create procedure 存储过程名(参数列表)

begin

一组sql语句

End

参数包含三部分：参数模式 参数名 参数类型 例：in stuname varchar(20)

参数模式：

① in ： 该参数可以作为输入，需要调用方传入值。

② out ：该参数可以做为输出，该参数可以作为返回值

③ inout ：该参数即可做输入又可做输出，该参数即需要传入值，又可作为返回值。

在存储过程中如果仅仅只有一句，begin end 可以省略，

存储过程中的每条sql语句都是用分号结尾，但存储过程的结尾可以使用 delimiter 重新设置。 Delimiter 结束标记 例： delimiter $

1. 调用语法

Call 存储过程名(实参列表)

in例：;

Delimiter $

Create procedure myp1(in beautyName varchar(20))

begin

declare boyName varchar(20); #声明变量

select b.name into boyName from boy b left join beauty be on b.id=be.boyfirend\_id where be.name=beautyName; #通过select 字段名 into from表；形式赋值

select boyName; #查看变量

end $

调用 call myp1(‘小昭’)$

out例：Delimiter $

Create procedure myp2(in beautyName varchar(20),out boyName varchar(20))

begin

select b.name into boyName from boy b left join beauty be on b.id=be.boyfirend\_id where be.name=beautyName; #通过select 字段名 into from表；形式赋值

end $

调用：【set @bName varchar(20)$】call myp2(‘小昭’,@bName)$

查看： select @bName$

Inout例：

Create procedure myp2(inout a int, inout b int)

begin

set a=a+1;

set b=b+2;

end $

调用：set @a:=0$ set @b:=0$ call myp3(@a,@b)$ #(inout 传入的必须是变量)

查看：select @a,@b#

1. 存储过程的删除：

Drop procedure 存储过程名； #一次只能删除一个

1. 存储过程的查看：

Show create procedure 存储过程名；

四、函数

1、创建

Delimiter 结束标记

Create function 函数名（参数列表） returns 返回值类型 #注：returns而不是return

Begin

函数体

End 结束标记

注：参数列表 包含两个部分 ：参数名 参数类型

函数体：肯定会有return语句，如果没有会报错。

使用delimiter设置结束表示。

1. 调用语法：

Select 函数名（参数列表）

函数：有且仅有一个返回值

五、流程控制结构

1、If函数

If（表达式1，表达式二，表达式三）； #1成立则返回表达式二结果，否则返回三结果

1. Case结构

情况1语法 ：（相当于java中switch 实现等值判断）

Case 变量/表达式/字段

When 要判断的自 then 返回值1或者语句1；

……

Else 要返回的值n或者语句n；

End case；

情况2语法：类似于java中多重if判断

Case

When 要判断的条件 then 返回值或者语句；

…….

Else 返回值或语句；

End case；

注： 可以做表达式，嵌套在其他语句中使用

做独立语句使用时，只能用在begin end 中。

1. if结构

if 条件 then 语句1；

elseif 条件2 then语句2；

…….

【else 语句n】

End if；

注：应用于begin end中