MySQL入门 SQL语言之十八：系统变量(全局变量、会话变量)，自定义变量(用户变量、局部变量)的使用

#变量

/\*

系统变量：

    全局变量

    会话变量

自定义变量：

    用户变量

    局部变量

\*/

#一、系统变量

/\*

说明：变量由系统提供，不是用户定义，属于服务器层面

使用的语法：

1、查看所有的系统变量

show global variables;

SHOW session VARIABLES;#session可以省略

2、查看满足条件的部分系统变量

show VARIABLES variables like '%char%';

3、查看指定的某个系统变量值

select @@global|【session】.系统变量名;

4、为某个系统变量赋值

方式一：

set global|【session】 系统变量名 = 值;

set autocommit = 0;

方式二：

set @@global|【session】.系统变量名 = 值;

注意：

如果是全局级别，则需要加global，如果是会话级别，则需要加session，如果不写，则默认session

\*/

#1》、全局变量

/\*

作用域：服务器每次启动将为所有的全局变量赋初始值，针对于所有会话（连接）有效，但是不能跨重启。

如果要想每次启动也修改，则需要修改配置文件。

\*/

#①查看所有的全局变量

SHOW GLOBAL VARIABLES;

#②查看部分的全局变量

SHOW GLOBAL VARIABLES LIKE '%char%';

#③查看指定的全局量的值

SELECT @@global.autocommit;

SELECT @@tx\_isolation;

#④为某个指定全局变量赋值

#方式一：

SET @@global.autocommit = 0;

#方式二：

SET GLOBAL autocommit = 0;

#2》、会话变量

/\*

作用域：仅仅针对于当前会话（连接）有效，并且有默认值

\*/

#①查看所有的会话变量;

SHOW SESSION VARIABLES;

SHOW  VARIABLES;

#②查看部分的会话变量

SHOW  VARIABLES LIKE '%char%';

SHOW SESSION VARIABLES  LIKE '%char%';

#③查看指定的某个会话变量

SELECT @@tx\_isolation;

SELECT @@session.tx\_isolation;

#④为某个会话变量赋值

#方式一

SET @@session.tx\_isolation = 'read-uncommitted';

#方式二：

SET SESSION tx\_isolation =  'read-committed';

#二、自定义变量

/\*

说明：变量是用户自定义的，不是由系统的

使用步骤

声明

赋值

使用（查看、比较、运算等）

\*/

#1、用户变量

/\*

作用域：针对于当前会话（连接）有效，同于会话变量的作用域

应用在任何地方，也就是begin end里面或begin end外边

\*/

赋值的操作符： = 或 :=

#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(更新用户变量的值)

方式一: 通过SET或SELECT

    SET @用户变量名 = 值;

    SET @用户变量名 := 值;

    SELECT @用户变量名 :=值;

方式二：通过SELECT INTO

    SELECT 字段 INTO @变量名

    FROM 表;

#③使用（查看用户变量名的值）

SELECT @用户变量名;

#2、局部变量

/\*

作用域：仅仅在定义它的begin end中有效

应用在begin end中的第一句话

\*/

#①声明

DECLARE 变量名 类型;

DECLARE 变量名 类型 DEFAULT 值;

#②赋值

方式一: 通过SET或SELECT

    SET 局部变量名 = 值;

    SET 局部变量名 := 值;

    SELECT @局部变量名 :=值;

方式二：通过SELECT INTO 局部变量名

    SELECT 字段

#③使用

SELECT 局部变量名;

#案例：

#声明并初始化

SET @name = 'john';

SET @name = 100;

SET @count = 1;

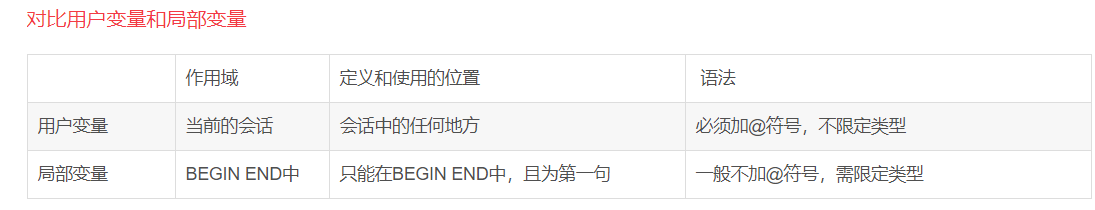
#赋值

SELECT COUNT(\*) INTO @count

FROM employees;

#查看

SELECT @count;



#案例：声明两个变量并赋初始值，就和，并打印

#1.用户变量

SET @m=1;

SET @n=2;

SET @sum = @m+@n;

SELECT @sum;

#2.局部变量

    DECLARE m INT DEFAULT 1;

    DECLARE n INT DEFAULT 2;

    DECLARE SUM INT;

    SET SUM = m + n;

    SELECT SUM;

#报错

---------------------

作者：不等风雨，只等你

来源：CSDN

原文：https://blog.csdn.net/qq\_34626097/article/details/86528466

版权声明：本文为博主原创文章，转载请附上博文链接！

说明：现在市面上定义变量的教程和书籍基本都放在存储过程上说明，但是存储过程上变量只能作用于begin...end块中，而普通的变量定义和使用都说的比较少，针对此类问题只能在官方文档中才能找到讲解。

**前言**

MySQL存储过程中，定义变量有两种方式：

1、使用set或select直接赋值，变量名以@开头

例如:

set @var=**1**;

可以在一个会话的任何地方声明，作用域是整个会话，称为用户变量。

2、以declare关键字声明的变量，只能在存储过程中使用，称为存储过程变量，例如：

declare var1 int default **0**;

主要用在存储过程中，或者是给存储传参数中。

两者的区别是：

在调用存储过程时，以declare声明的变量都会被初始化为null。而会话变量（即@开头的变量）则不会被再初始化，在一个会话内，只须初始化一次，之后在会话内都是对上一次计算的结果，就相当于在是这个会话内的全局变量。

**主体内容:系统变量(*全局变量*、*会话变量*)，自定义变量(*局部变量、用户变量*)**

* 局部变量=存储过程变量
* 用户变量
* 会话变量
* 全局变量

会话变量和全局变量叫系统变量。

**一、局部变量，只在当前begin/end代码块中有效**

局部变量一般用在sql语句块中，比如存储过程的begin/end。其作用域仅限于该语句块，在该语句块执行完毕后，局部变量就消失了。declare语句专门用于定义局部变量，可以使用default来说明默认值。set语句是设置不同类型的变量，包括会话变量和全局变量。   
局部变量定义语法形式

declare var\_name [, var\_name]... data\_type [ DEFAULT value ];

例如在begin/end语句块中添加如下一段语句，接受函数传进来的a/b变量然后相加，通过set语句赋值给c变量。

set语句语法形式set var\_name=expr [, var\_name=expr]...; set语句既可以用于局部变量的赋值，也可以用于用户变量的申明并赋值。

declare c int default **0**;

set c=a+b;

select c as C;

或者用select …. into…形式赋值

select into 语句句式：select col\_name[,...] into var\_name[,...] table\_expr [where...];

例子：

[复制代码](javascript:void(0);)

declare v\_employee\_name varchar(**100**);

declare v\_employee\_salary decimal(**8**,**4**);

select employee\_name, employee\_salary

into v\_employee\_name, v\_employee\_salary

from employees

where employee\_id=**1**;

[复制代码](javascript:void(0);)

**二、用户变量，在客户端链接到数据库实例整个过程中用户变量都是有效的。**

MySQL中用户变量不用事先申明，在用的时候直接用“@变量名”使用就可以了。

第一种用法：set @num=1; 或set @num:=1; //这里要使用set语句创建并初始化变量，直接使用@num变量

第二种用法：select @num:=1; 或 select @num:=字段名 from 表名 where ……，

**select语句一般用来输出用户变量，比如select @变量名，用于输出数据源不是表格的数据。**

注意上面两种赋值符号，使用set时可以用“=”或“:=”，但是使用select时必须用“:=赋值”

**用户变量与数据库连接**有关，在连接中声明的变量，在存储过程中创建了用户变量后一直到数据库实例接断开的时候，变量就会消失。

在此连接中声明的变量无法在另一连接中使用。

用户变量的变量名的形式为@varname的形式。

名字必须以@开头。

声明变量的时候需要使用set语句，比如下面的语句声明了一个名为@a的变量。

set @a = 1;

**声明一个名为@a的变量，并将它赋值为1，MySQL里面的变量是不严格限制数据类型的，它的数据类型根据你赋给它的值而随时变化** 。（SQL SERVER中使用declare语句声明变量，且严格限制数据类型。）

我们还可以使用select语句为变量赋值 。

比如：

set @name = '';

select @name:=password from user limit **0**,**1**;  
#从数据表中获取一条记录password字段的值给@name变量。在执行后输出到查询结果集上面。

（注意等于号前面有一个冒号，后面的limit 0,1是用来限制返回结果的，表示可以是0或1个。相当于SQL SERVER里面的top 1）

如果直接写：select @name:=password from user;

如果这个查询返回多个值的话，那@name变量的值就是最后一条记录的password字段的值 。

用户变量可以作用于当前整个连接，但当当前连接断开后，其所定义的用户变量都会消失。

用户变量使用如下（我们无须使用declare关键字对用户变量进行定义，可以直接这样使用）定义，变量名必须以@开始：

#定义

select @变量名 或者 select @变量名:= 字段名 from 表名 where 过滤语句;

set @变量名;

#赋值 @num为变量名，value为值

set @num=value;或select @num:=value;

对用户变量赋值有两种方式，一种是直接用”=”号，另一种是用”:=”号。其区别在于使用set命令对用户变量进行赋值时，两种方式都可以使用；当使用select语句对用户变量进行赋值时，只能使用”:=”方式，因为在select语句中，”=”号declare语句专门用于定义局部变量。set语句是设置不同类型的变量，包括会话变量和全局变量。

例如：

[复制代码](javascript:void(0);)

begin

#Routine body goes here...

#select c as c;

declare c int default **0**;

set @var1=**143**; #定义一个用户变量，并初始化为143

set @var2=**34**;

set c=a+b;

set @d=c;

select @sum:=(@var1+@var2) as sum, @dif:=(@var1-@var2) as dif, @d as C;#使用用户变量。@var1表示变量名

set c=**100**;

select c as CA;

end

#在查询中执行下面语句段

call `order`(**12**,**13**); #执行上面定义的存储过程

select @var1; #看定义的用户变量在存储过程执行完后，是否还可以输出，结果是可以输出用户变量@var1,@var2两个变量的。

select @var2;

[复制代码](javascript:void(0);)

在执行完order存储过程后，在存储过程中新建的var1，var2用户变量还是可以用select语句输出的，但是存储过程里面定义的局部变量c不能识别。

**系统变量**：

系统变量又分为全局变量与会话变量。

全局变量在MySQL启动的时候由服务器自动将它们初始化为默认值，这些默认值可以通过更改my.ini这个文件来更改。

会话变量在每次建立一个新的连接的时候，由MySQL来初始化。MySQL会将当前所有全局变量的值复制一份。来做为会话变量。

（也就是说，如果在建立会话以后，没有手动更改过会话变量与全局变量的值，那所有这些变量的值都是一样的。）

全局变量与会话变量的区别就在于，对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，只会影响到当前的会话（也就是当前的数据库连接）。

我们可以利用

show session variables;

语句将所有的会话变量输出（可以简写为show variables，没有指定是输出全局变量还是会话变量的话，默认就输出会话变量。）如果想输出所有全局变量：

show global variables;

有些系统变量的值是可以利用语句来动态进行更改的，但是有些系统变量的值却是只读的。

对于那些可以更改的系统变量，我们可以利用set语句进行更改。

**系统变量在变量名前面有两个@；**

如果想要更改**会话变量**的值，利用语句：

set session varname = value;

或者

set **@@session**.varname = value;

比如：

mysql> set session sort\_buffer\_size = **40000**;

Query OK, **0** rows affected(**0.00** sec)

用select **@@sort\_buffer\_size**;输出看更改后的值是什么。

如果想要更改全局变量的值，将session改成global：

set global sort\_buffer\_size = **40000**;

set **@@global**.sort\_buffer\_size = **40000**;

不过要想更改全局变量的值，需要拥有super权限 。

（注意，root只是一个内置的账号，而不是一种权限 ，这个账号拥有了MySQL数据库里的所有权限。任何账号只要它拥有了名为super的这个权限，就可以更改全局变量的值，正如任何用户只要拥有file权限就可以调用load\_file或者into outfile，into dumpfile，load data infile一样。）

利用select语句我们可以**查询单个会话变量或者全局变量**的值：

select **@@session**.sort\_buffer\_size

select **@@global**.sort\_buffer\_size

select **@@global**.tmpdir

凡是上面提到的session，都可以用local这个关键字来代替。

比如：

select **@@local**.sort\_buffer\_size

local是session的近义词。

无论是在设置系统变量还是查询系统变量值的时候，只要没有指定到底是全局变量还是会话变量。都当做会话变量来处理。

比如：

set **@@sort\_buffer\_size** = **50000**;

select **@@sort\_buffer\_size**;

上面都没有指定是blobal还是session，所以全部当做session处理。

**三、会话变量**

服务器为每个连接的客户端维护一系列会话变量。在客户端连接数据库实例时，使用相应全局变量的当前值对客户端的会话变量进行初始化。设置会话变量不需要特殊权限，但客户端只能更改自己的会话变量，而不能更改其它客户端的会话变量。**会话变量的作用域与用户变量一样，仅限于当前连接。当当前连接断开后，其设置的所有会话变量均失效。**

设置会话变量有如下三种方式更改会话变量的值：

set session var\_name = value;

set **@@session**.var\_name = value;

set var\_name = value; #缺省session关键字默认认为是session

查看所有的会话变量

show session variables;

查看一个会话变量也有如下三种方式：

select **@@var\_name**;

select **@@session**.var\_name;

show session variables like "%var%";

凡是上面提到的session，都可以用local这个关键字来代替。

比如：

select **@@local**.sort\_buffer\_size

local是session的近义词。

**四、全局变量**

全局变量影响服务器整体操作。当服务器启动时，它将所有全局变量初始化为默认值。这些默认值可以在选项文件中或在命令行中指定的选项进行更改。要想更改全局变量，必须具有super权限。全局变量作用于server的整个生命周期，但是不能跨重启。即重启后所有设置的全局变量均失效。要想让全局变量重启后继续生效，需要更改相应的配置文件。

要设置一个全局变量，有如下两种方式：

set global var\_name = value; //注意：此处的global不能省略。根据手册，set命令设置变量时若不指定GLOBAL、SESSION或者LOCAL，默认使用SESSION

set **@@global**.var\_name = value; //同上

查看所有的全局变量

show global variables;

要想查看一个全局变量，有如下两种方式：

select **@@global**.var\_name;

show global variables like “%var%”;

参考：

<http://blog.csdn.net/JQ_AK47/article/details/52087484>（以上内容转自此篇文章）

<http://blog.csdn.net/seteor/article/details/17682551>

<http://blog.csdn.net/jssg_tzw/article/details/39703889?utm_source=tuicool&utm_medium=referral>

<http://www.cnblogs.com/qixuejia/archive/2010/12/21/1913203.html>

<http://blog.163.com/longsu2010@yeah/blog/static/173612348201162595425697/>

==>如有问题，请联系我：easonjim#163.com，或者下方发表评论。<==

分类: [数据库技术-[MySQL/Oracle/SQLServer]](https://www.cnblogs.com/EasonJim/category/475875.html)

标签: [mysql](https://www.cnblogs.com/EasonJim/tag/mysql/)