



武汉生物工程学院

# 应用数据库 (MySQL)

主讲教师：朱 华

Q Q 号：369620942

签到方式：微助教

联系电话：13135670337

课堂编号：DJ13(?)



## MySQL函数

为了便于MySQL代码维护，以及提高MySQL代码的重用性，MySQL开发人员经常将频繁使用的业务逻辑封装成**存储程序**。

MySQL的存储程序分为四类：**函数**、**触发器**、**存储过程**以及**事件**。通过本章学习为将来的数据库编程奠定坚实的基础。





## 常量

按照MySQL的数据类型进行划分，常量可分为数值型常量、十六进制常量、二进制常量、字符串常量、日期型常量、布尔值、空值等。

### 1、数值型常量

1221、0、-32、294.42、0.148、  
-32032.6809等。





## 常量

### 2、十六进制常量

表示方法①0x0a=10、0xffff=65535

十六进制数字不区分大小写。

表示方法②X '41' =A、

X '4D7953514C' =MySQL





## 常量

```
mysql> select hex('mysql');
+-----+
| hex('mysql') |
+-----+
| 6D7973716C   |
+-----+
1 row in set (0.03 sec)

mysql> select hex('MYSQL');
+-----+
| hex('MYSQL') |
+-----+
| 4D5953514C   |
+-----+
1 row in set (0.00 sec)
```





## 常量

十六进制常量的默认类型是字符串。如果要输出数值，可以使用转换函数cast (.....as Unsigned)，或者在十六进制数后面“+0”。例sql语句如下：

```
Select x' 41', 0x41+0, cast (0x41 as unsigned), hex (41), x' 4d7953514c' ;
```





## 常量

十六进制常量的默认类型是字符串。如果要输出数值，可以使用转换函数`cast (.....as Unsigned)`，或者在十六进制数后面“+0”。例sql语句如下：

```
Select x'41', 0x41+0, cast(0x41 as unsigned), hex(41), x'4d7953514c' ;
```

```
mysql> Select x'41', 0x41+0, cast(0x41 as unsigned), hex(41), x'4d7953514c' ;
+-----+-----+-----+-----+-----+
| x'41' | 0x41+0 | cast(0x41 as unsigned) | hex(41) | x'4d7953514c' |
+-----+-----+-----+-----+-----+
| A     | 65     | 65 | 29     | MySQL         |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```



## 常量

### 3、二进制常量

二进制常量由数字“0”和“1”组成。二进制常量的表示方法为“b+’二进制字符串’”。输出字符型的方法与十六进制常量相同。例sql语句如下：

```
Select b'111101', b'1', b'11', b'11'+0,  
cast(b'11' as unsigned), bin(b'11'+0);
```







## 常量

### 3、二进制常量

二进制常量由数字“0”和“1”组成。二进制常量的表示方法为“b+’二进制字符串’”。输出字符型的方法与十六进制常量相同。例sql语句如下：

```
Select b'111101',b'1',b'11',b'11'+0,  
cast(b'11' as unsigned),bin(b'11'+0);
```

```
mysql> Select b'111101',b'1',b'11',b'11'+0,  
-> cast(b'11' as unsigned),bin(b'11'+0);  
+-----+-----+-----+-----+-----+-----+  
| b'111101' | b'1' | b'11' | b'11'+0 | cast(b'11' as unsigned) | bin(b'11'+0) |  
+-----+-----+-----+-----+-----+-----+  
| =         | 1    | 11    | 3       | 3       | 11          |  
+-----+-----+-----+-----+-----+-----+  
1 row in set (0.02 sec)
```



## 常量

### 4、字符串常量

字符串常量是指加单引号或双引号的字符序列。字符串中使用的字符，有一些被用来表示了特定的含义，如'、%、\_等，这些字符不同于字符原有的含义，即为**转义字符**，转义字符常以“\”开头。





## 常量

```
mysql> SELECT 'This\nIs\nFour\nLines';  
+-----+  
| This  
Is  
Four  
Lines |  
+-----+  
| This  
Is  
Four  
Lines |  
+-----+  
1 row in set (0.00 sec)
```



## 常量

```
select 'I\'m a teacher' as col1,  
"you're a stude\nt" as col2;
```

由于大多编程语言（例如Java、C等）使用双引号表示字符串，为了便于区分，在MySQL数据库中推荐使用单引号表示字符串。





## 常量

```
mysql> select 'I\'m a teacher' as col1,  
-> "you're a stude\nt" as col2;  
+-----+-----+  
| col1          | col2          |  
+-----+-----+  
| I'm a teacher | you're a stude  
t |  
+-----+-----+  
1 row in set (0.00 sec)
```





## 常量

### 5、日期时间常量

如 “20:31:00”、“2019-11-11 11:11:11”

日期时间常量的值必须符合日期、时间标准

### 6、布尔值

true:1和false:0

### 7、null值

表示“没有值”，不同于“0”





## 变量

MySQL数据库中的变量分为系统变量（以@@开头）和用户自定义变量。

### 系统变量

系统变量分为全局系统变量（GLOBAL SYSTEM VARIABLE）和会话系统变量（SESSION SYSTEM VARIABLE）。





## 全局系统变量

使用“**show global variables;**”命令可查看mysql服务器中所有的全局系统变量。

当MySQL服务器启动的时候，全局系统变量被初始化为默认值，并且**应用于每个启动的会话**。要想修改全局系统变量的值可以在选项文件中或在命令行中对指定的选项进行修改。**命令行中的修改重启后所有设置的全局系统变量均失效。**

例6.3：将全局系统变量sort\_buffer\_size的值改为25000。Sql语句如下：

```
set @@global.sort_buffer_size=25000;
```







## 会话系统变量

会话系统变量只适用于当前的会话。大多数会话系统变量的名字和全局系统变量的名字相同。

例6.5：基于当前会话，把系统变量 `sql_select_limit` 的值设置为10。这个变量决定了 `select` 语句的结果集中的最大行数。Sql语句如下：

```
set @@session.sql_select_limit=10;
```

```
set @@character_set_server=gbk;
```

设置mysql默认字符集。





## 用户自定义变量

用户自定义变量分为用户会话变量（以@开头）和局部变量（不以@开头）。

### 局部变量

局部变量必须定义在存储程序中（例如函数、触发器、存储过程以及事件中），一般用在sql语句块中。使用之前用declare命令定义局部变量及其数据类型，例如“declare c Char(20)”。





## 用户自定义变量

局部变量主要用于下面三种场合：

场合一：局部变量定义在存储程序的**begin**和**end**语句块之间；

场合二：局部变量作为存储过程或者函数的参数使用，**需要指定参数的数据类型**；

场合三：局部变量也可以用在sql语句中。





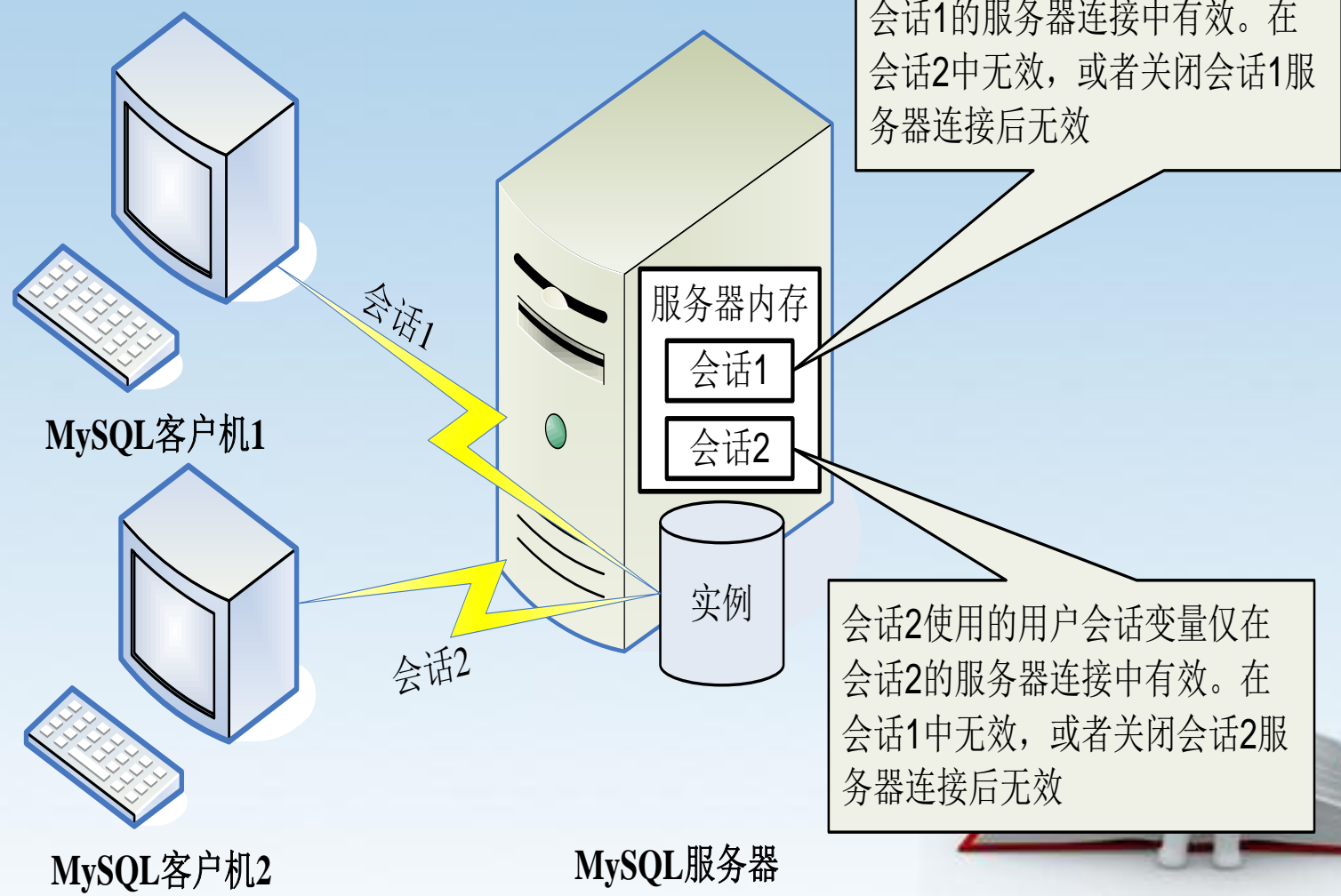
## 用户自定义变量

### 用户会话变量

用户会话变量是MySQL客户机与数据库服务器在会话期间使用的变量，作用范围只和客户机相关。具体如图示：



## 用户自定义变量





## 用户自定义变量

系统会话变量与用户会话变量的共同之处在于：变量名大小写不敏感。系统会话变量与用户会话变量的区别在于：

1、用户会话变量一般以一个“@”开头；系统会话变量以两个“@”开头。

2、系统会话变量无需定义可以直接使用；用户会话变量需要定义与赋值，一般情况下，用户会话变量的定义和赋值会同时进行。





## 用户自定义变量

用户会话变量的定义与赋值有三种方法：

方法一：使用set命令定义用户会话变量，并为其赋值，语法格式为：

```
set @user_variable1=值1  
[, @user_variable2=值2, ...];
```





## 用户自定义变量

方法二：使用select语句定义用户会话变量，  
并为其赋值，语法格式为：

```
select @user_variable1:=值1  
[, user_variable2:= 值2 , ...];
```

使用“:=”赋值，因为select语句中“=”是  
比较运算符。







## 用户自定义变量

方法三：使用select……into语句定义用户会话变量，并为其赋值，语法格式为：

```
select 值1 into @user_variable1, 值2  
into @user_variable2, …;
```





## 用户自定义变量

例6.6 创建用户变量name并赋值为“王林”；创建用户变量user1并赋值为1，user2赋值为2，user3赋值为3；创建用户变量user4，它的值为user3的值加1。Sql语句如下：

```
select '王林' into @name;  
set @user1=1, @user2=2, @user3=3;  
select @user4:=@user3+1;
```

使用select语句输出各个变量的值。





## 用户自定义变量

```
mysql> select '王林' into @name;
Query OK, 1 row affected (0.04 sec)

mysql>      set @user1=1,@user2=2,@user3=3;
Query OK, 0 rows affected (0.00 sec)

mysql>      select @user4:=@user3+1;
+-----+
| @user4:=@user3+1 |
+-----+
|                4 |
+-----+
1 row in set (0.00 sec)

mysql> select @name,@user1,@user2,@user3,@user4;
+-----+-----+-----+-----+-----+
| @name | @user1 | @user2 | @user3 | @user4 |
+-----+-----+-----+-----+-----+
| 王林  |      1 |      2 |      3 |      4 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```



## 用户自定义变量

例6.7 可以将表达式赋值给变量。Sql语句如下：

```
select @t2:=(@t2:=2)+5 as t2;
```

```
mysql> select @t2:=(@t2:=2)+5 as t2;  
+-----+  
| t2 |  
+-----+  
| 7 |  
+-----+  
1 row in set (0.00 sec)
```



## 用户自定义变量

例6.8 查询学生信息表student中学号为“01640401”学生的年龄并赋值给变量@a，查看变量的值。Sql语句如下：

```
set @a=(select year(now())-year(出生日期)
from student where 学号='01640401');
select @a;
```





## 用户自定义变量

```
mysql> set @a=(select year(now())-year(出生日期)
-> from student where 学号='01640401');
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select @a;
+-----+
| @a    |
+-----+
| 17    |
+-----+
1 row in set (0.00 sec)
```





## 局部变量与用户自定义变量的区别

1、用户会话变量名以“@”开头，而局部变量名前面没有“@”符号；

2、局部变量使用declare命令定义（存储过程参数、函数参数除外），定义时必须指定局部变量的数据类型；局部变量定义后，才可以使用set命令或select语句为其赋值，Declare命令尽量写在begin...end语句块中。用户会话变量使用set命令或select语句定义并进行赋值，定义用户会话变量时无须指定数据类型。





## 局部变量与用户自定义变量的区别

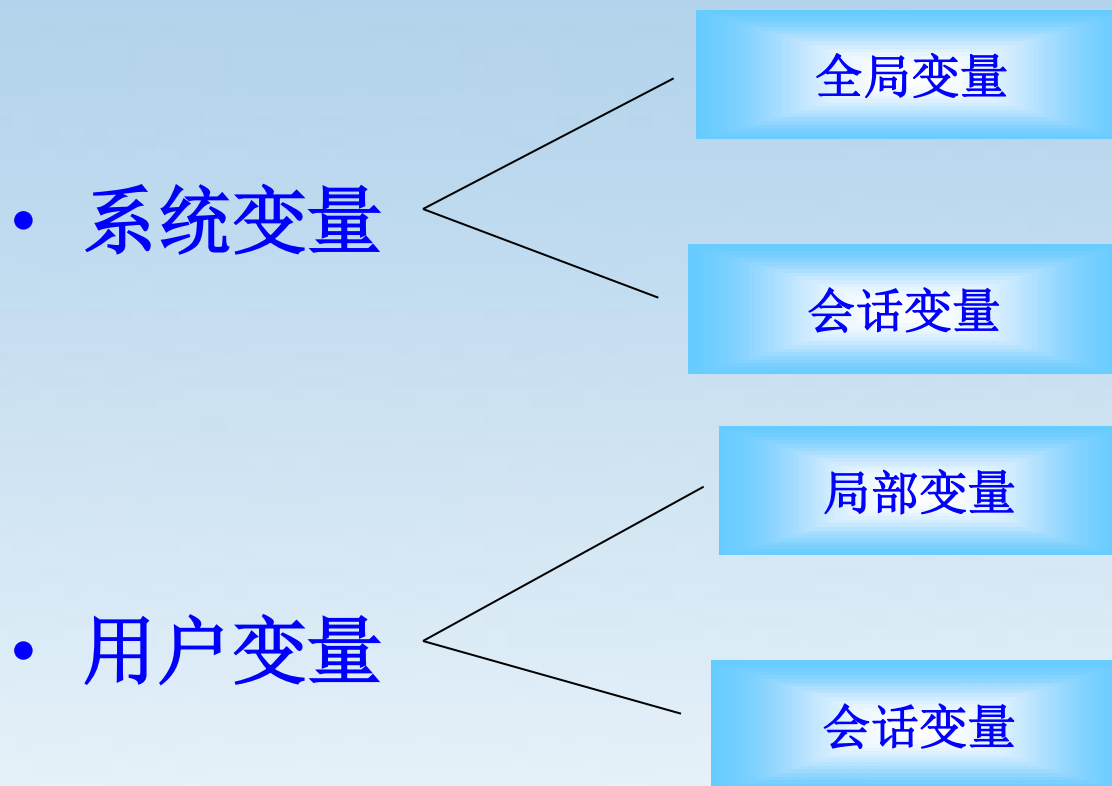
3、用户会话变量的作用范围与生存周期大于局部变量。局部变量如果作为存储过程或者函数的参数，此时在整个存储过程或函数内有效；如果定义在存储程序的begin...end语句块中，此时仅在语句块中有效。用户会话变量在本次会话期间一直有效。

4、如果局部变量嵌入到sql语句中，由于局部变量名前没有“@”符号，这就要求局部变量名不能与表字段名相同，否则将出现无法预期的后果。





# 变量



变量分为系统变量（以@@开头），用户自定义变量分为用户会话变量（以@开头）以及局部变量（不以@开头）。





## 运算符

- 1、算术运算符
- 2、比较运算符
- 3、逻辑运算符

在数据查询中已讲过，大家还记得吗？

### 4、位运算符

位运算符对二进制数据进行运算。

&：按位与    |：按位或    ~：按位取反

^：按位异或    <<：按位左移    >>：按位右移





## 运算符的优先级

运算符
=（赋值运算），:=
, OR
XOR
&&, AND
NOT
BETWEEN, CASE, WHEN, THEN, ELSE
=（比较运算），<=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP, IN
&
<<, >>
-, +
*, /(DIV), %(MOD)
^
-（负号），~（位反转）
!



## 变量的运用

```
Set @a=3;  
Set @b=4;  
Set @a=@a+@b;  
Select @a;
```

```
Set @a=3;  
Set @b=4;  
Set @a=@a+@b;  
Set @b=@a+@b;  
Select @a,@b;
```





## 变量的运用

```
mysql> Set @a=3;
Query OK, 0 rows affected (0.00 sec)

mysql> Set @b=4;
Query OK, 0 rows affected (0.00 sec)

mysql> Set @a=@a+@b;
Query OK, 0 rows affected (0.05 sec)

mysql> Select @a;
+-----+
| @a    |
+-----+
|      7 |
+-----+
1 row in set (0.00 sec)
```





## 变量的运用

```
mysql> Set @a=3;
Query OK, 0 rows affected (0.00 sec)

mysql> Set @b=4;
Query OK, 0 rows affected (0.00 sec)

mysql> Set @a=@a+@b;
Query OK, 0 rows affected (0.00 sec)

mysql> Set @b=@a+@b;
Query OK, 0 rows affected (0.00 sec)

mysql> Select @a,@b;
+-----+-----+
| @a    | @b    |
+-----+-----+
|      7 |     11 |
+-----+-----+
1 row in set (0.00 sec)
```

