



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(?)



自定义函数

函数存储着一系列sql语句，作为数据库的一个对象存储到数据库服务器上。调用函数就是一次性执行这些语句，函数可以封装sql语句，减少语句重复编写工作，为数据库编程人员提供方便。





创建自定义函数

创建自定义函数时，数据库开发人员需提供**函数名**、**函数的参数**、**函数体**（一系列的操作）以及**返回值**等信息。创建自定义函数的语法格式为：

```
create function 函数名 (参数1, 参数2, ...)  
returns 返回值的数据类型
```

[函数选项]

```
begin  
    函数体;  
    return 语句;  
end;
```



```
language sql  
| [not] deterministic  
| { contains sql | no sql |  
reads sql data | modifies sql  
data }  
| sql security { definer | invoker }  
| comment '注释'
```



创建自定义函数

说明：

自定义函数是数据库的对象，创建自定义函数时，**需要打开数据库**。

同一个数据库内，自定义函数名不能与已有的函数名（包括系统函数）重复，函数名不能是关键字。**建议在自定义函数名添加前缀“_fn”或者后缀“fn_”。**





创建自定义函数

说明：

函数的参数无须使用declare命令定义，但它仍然是**局部变量**，且**必须定义参数的数据类型**，即使没有参数，“**()**”也不能省略，这是的函数为空参数函数。

函数必须指定**返回值数据类型**，且需与**return语句中的返回值的**数据类型一致（如果是字符串数据类型，长度可以不同）





创建自定义函数

函数选项说明：

language sql: 默认选项，用于说明函数体使用SQL语言编写。

deterministic (确定性)

当函数返回不确定值时，该选项是为了防止“复制”时的不一致性。如果函数总是对同样的输入参数产生同样的结果，则被认为它是“确定的”，否则就是“不确定”的。

例如函数返回系统当前的时间，返回值是不确定的。

如果既没有给定deterministic也没有给定not deterministic, 默认的就是not deterministic。



创建自定义函数

函数选项说明：

`contains sql`：表示函数体中**不包含读或写数据的语句**。

`no sql`：表示函数体中**不包含SQL语句**。

`reads sql data`：表示函数体中**包含select查询语句，但不包含更新语句**。





创建自定义函数

函数选项说明：

modifies sql data: 表示函数体**包含更新语句**。

sql security: 用于指定函数的**执行许可**。

definer: 表示该函数只能由**创建者调用**。

invoker: 表示该函数可以被**其他数据库用户调用**。

comment: 为函数添加功能说明等**注释信息**。





创建自定义函数

例6.9 创建自定义函数，根据学号查找学生的姓名。Sql语句如下：

```
delimiter $$  
create function sname_fn(a char(10))  
returns char(10)  
reads sql data  
begin  
declare s char(10);  
select 姓名 into s from student where  
学号=a;  
return s;  
end; $$
```





创建自定义函数

例6.10 创建自定义函数，根据学号查找学生所在的班级。Sql语句如下：

```
delimiter $$  
create function sclass_fn(a char(10))  
returns char(10)  
reads sql data  
begin  
declare s char(10);  
select 班级名称 into s from course  
where 班级编号=(select 班级编号 from student  
where 学号=a);  
return s;  
end; $$
```





创建自定义函数

例6.11 创建自定义函数，根据班级名称统计该班级的学生总人数。Sql语句如下：

```
delimiter $$  
create function cn_fn(a char(20))  
returns int  
reads sql data  
begin  
declare s int;  
select count(学号) into s from classes  
join student on classes.班级编号=student.班  
级编号 where 班级名称=a;  
return s;  
end; $$
```





创建自定义函数

例6.12 创建名字为row_no_cn()的函数，函数功能是为查询结果集添加行号。Sql语句如下：

```
delimiter $$  
create function row_no_fn()  
returns int  
no sql  
begin  
set @row_no=@row_no+1;  
return @row_no;  
end; $$
```

在自定义函数时，不是所有的函数都需要参数，在使用函数时不需要输入任何数据，则创建函数时不需要定义参数，即空参数函数。





自定义函数的调用

调用自定义函数与调用系统函数的方法一样，使用select语句。

例6.9 查询学号为01640406的学生姓名。

```
select sname_fn('01640406');$$
```

例6.12 在表中使用row_no函数生成行号。

```
set @row_no=0;  
select row_no_fn(), 学号, 姓名 from  
student;$$
```





自定义函数的调用

```
mysql> select row_no_fn(),学号,姓名 from student;$$
```

| row_no_fn() | 学号 | 姓名 |
|-------------|----------|----|
| NULL | 01640401 | 浅浅 |
| NULL | 01640402 | 猴子 |
| NULL | 01640403 | 魔兽 |
| NULL | 01640404 | 英雄 |
| NULL | 01640405 | 火王 |
| NULL | 01640406 | 马浅 |
| NULL | 01640407 | 田七 |

```
7 rows in set (0.02 sec)
```





自定义函数的调用

例6.13 调用例6.11的cn_fn函数，返回“19信管1班”的学生人数。

```
select * from classes;
```

```
select cn_fn('19信息管理1班');$$
```





自定义函数的调用

```
mysql> select * from classes;  
->      select cn_fn('19信息管理1班');$$
```

| 班级编号 | 班级名称 | 学院编号 |
|------|--------------|------|
| 01 | 19计算机科学与技术1班 | 3 |
| 02 | 19计算机科学与技术2班 | 3 |
| 03 | 19计算机科学与技术3班 | 3 |
| 04 | 19信息管理1班 | 4 |
| 05 | 19机电1班 | 1 |
| 06 | 19机电2班 | 1 |
| 07 | 19生工1班 | 5 |

7 rows in set (0.00 sec)

```
+-----+  
| cn_fn('19信息管理1班') |  
+-----+  
| 1 |  
+-----+  
1 row in set (0.06 sec)
```




自定义函数的调用

例6.14 函数创建与调用：在图书馆销售管理数据中心，根据书店编号查询该书店图书的上架数量。Sql语句如下：

```
delimiter $$  
create function s_fn(n char(10))  
returns char(10)  
reads sql data  
begin  
declare m char(10);  
set m=(select sum(上架数量) from onsale  
where 书店id=n);  
return m;  
end; $$
```





自定义函数的调用

```
mysql> delimiter $$
mysql> create function s_fn(n char(10))
-> returns char(10)
-> reads sql data
-> begin
-> declare m char(10);
-> set m=(select sum(上架数量) from onsale where 书店id=n);
-> return m;
-> end; $$
Query OK, 0 rows affected (0.00 sec)

mysql> select s_fn(3),s_fn(4);$$
+-----+-----+
| s_fn(3) | s_fn(4) |
+-----+-----+
| 1182    | 1071    |
+-----+-----+
1 row in set (0.02 sec)
```



自定义函数的删除

删除函数使用sql语句“drop function 函数名;”。例如删除s_fn()函数可以使用“drop function s_fn;”





自定义函数的维护

函数的维护主要是查看函数的定义，常用的方法有以下四种：

1、查看当前数据库中所有的自定义函数信息，可以使用MySQL命令“`show function status;`”。

2、查看指定数据库（例如choose数据库）中的所有自定义函数名，可以使用下面的SQL语句：

```
select name from mysql.proc where db =  
'choose' and type = 'function';
```





自定义函数的维护

3、使用MySQL命令“**show create function 函数名;**”可以查看指定函数名的详细信息。

4、函数的信息都保存在information_schema数据库中的routines表中，可以使用select语句检索routines表，查询函数的相关信息。

```
select * from  
information_schema.routines where  
routine_name='get_name_fn' \G
```





自定义函数的逻辑

函数名

定义a, b 数据类型

函数返回值的类型

执行功能

$$c = a + b$$

返回结果





自定义函数练习题

现有学生信息表student，表中字段有学号(char (10)), 姓名(char (10)), 性别(char (1)), 出生日期(date), 入学成绩(decimal (5, 2)), 学院(char (20))请按要求完成下列练习。

- 1、在student表中通过学号查询学生性别;
- 2、在student表中通过学号查询学生年龄;
- 3、在student表中通过学号查询学生的入学成绩排名;





自定义函数小结

本讲介绍关于自定义函数的有关操作，主要包括自定义函数的**创建（重点是理解是掌握语法格式及函数选项）、查看、删除**等操作。函数的特点：
函数可以没有参数，即**空参数函数**；
函数必须有**返回值**，而且**只能有一个值**；
函数体中**不能直接输出结果**，需将结果**复制给变量**。

通过对本部分内容的学习，我们应掌握自定义函数的基本概念，而且还能编写一些简单的自定义函数。

