# day11 MySQL

## 一.内容简介

1. 存储过程(函数);
2. 触发器;
3. MySQL权限问题;
4. MySQL的备份与恢复.

## 二.存储过程(函数)

### 1.概念:

带有逻辑判断的sql语句,就是所谓的存储过程(函数).

也就是说可以给sql语句添加if....else...while等语句.

### 2.特点:

①.执行效率高:存储过程(函数)是在MySQL的服务器端来执行的语句,效率就比在Java中结合SQL判断效率高.

②.可移植性差:不同的数据库软件,支持的存储过程(函数)的语法不同,所以不同的数据库软件之间,存储过程不能直接移植.

### 3.存储过程语法

delimiter $

create procedure 存储过程名(参数)

begin

.....具体的代码

end $

delimiter:声明,$是存储过程开始与结束的标记.

### 4.存储函数语法

delimiter $

create function 存储函数名(参数) return 返回值类型

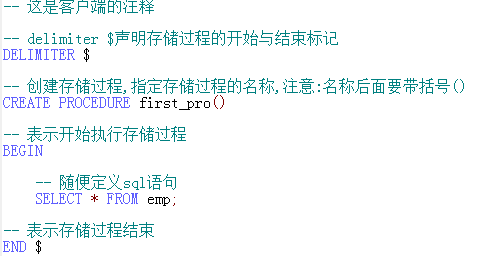
begin

...具体的代码

end $

### 具体案例

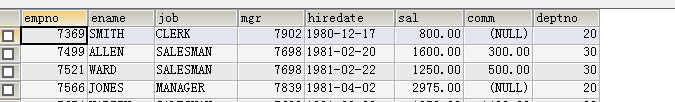
#### .定义不带参数的存储过程



**调用存储过程:**



**执行结果:**



#### .存储过程的参数类型

输入参数:IN

IN id INT:

IN:表示要输入一个参数;

id:参数名称;

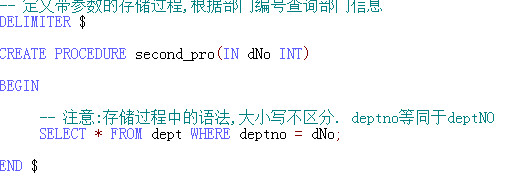
INT:参数类型.

输出参数:OUT

OUT msg char(20)

输入输出 :INOUT

#### .定义一个带有输入参数的存储过程



调用存储过程:



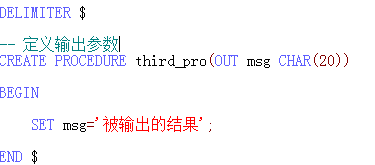
结果:



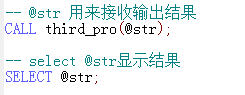
#### .删除存储过程



#### .带有输出参数的存储过程

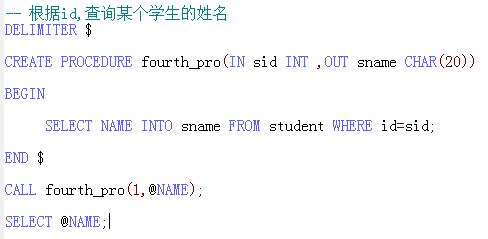


执行结果:



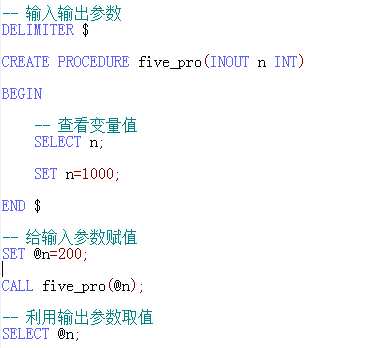


#### .带有输入和输出参数的例子





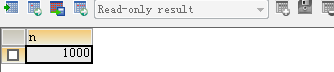
#### .输入输出参数INOUT



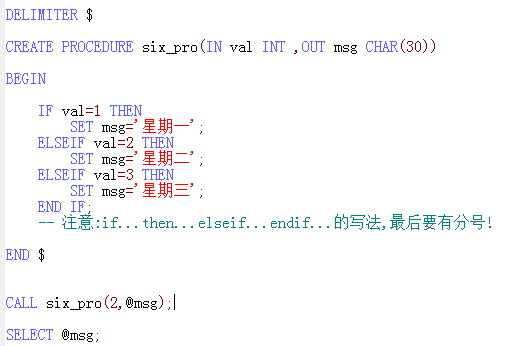
结果一:

n=200

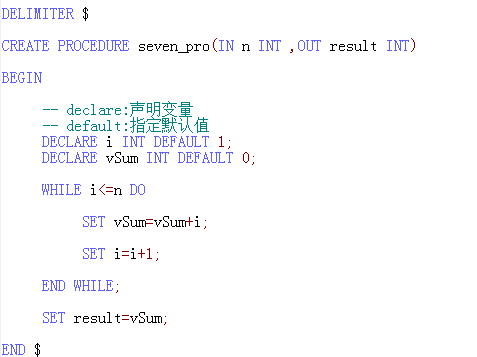
结果二:



#### .存储过程中使用if...then...elseif...end if;语句.



#### .存储过程中使用循环while





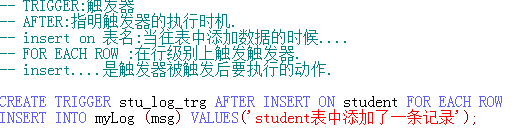
## 三.触发器

### 1.概念:

在操作一张表的时候,与此同时要触发另外的一些行为和动作,这样的操作就可以使用触发器.比如当操作一个表的时候,触发另一表来进行记录操作.

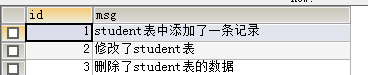
比如有个员工表,当往员工表中插入/update/delete等操作的时候,与此同时往另外一个log表插入一条记录.

### 2.语法:





结果:



## 四.视图View

### 1.概念:

视图也是一个”表”,只不过是一个”虚拟的表”.

视图是基于SQL(select)语句的结果集的可视化的表.

### 2.作用:

提高查询的效率,一般只用于查询.

### 3.语法

create View view\_name as

select 列名 from 表名

where....

注意:

as后面的查询语句,可以是我们之前的任意一种select语句.

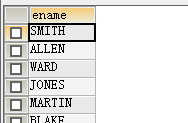
单表,多表查询,多表连接等都可以.

案例:创建一个视图



查询视图:





## 五.MySQL权限问题

### 1.mysql数据库

存放MySQL自身的配置信息.

User表:用来存储MySQL的用户信息的表.

### 2.password()函数

将密码进行md5加密.

案例:





3.往user表中添加一个用户.



**注意:**

**当对user表进行了insert/update/delete等操作后,必须刷新权限,否则会导致以下错误!**





**错误信息:**



### 3.grant命令

**案例:**

**给localhost服务器中的tom分配查询权利,而且tom只能查询db01数据库中的emp表.**



**错误信息:**



## 六.MySQL数据库的备份和恢复

### 1.数据库备份



**注意:**

#### .以管理员身份打开命名行;

#### .整个的备份语句后面不要带分号;

### 2.数据库恢复



**注意:**

#### .恢复数据库用的命名是mysql,不是mysqldump!

#### ②.恢复数据库之前,要手动的在数据库的客户端中建立一个同名的空白数据库.

## 七.数据库设计规范

数据库设计的3大范式

### 第一范式:

**表中的每个字段都必须是不可再分割的独立单元.**

student-----🡪name 不符合第一范式!

student--🡪old\_name 符合第一范式!

### 第二范式:

**在第一范式的基础之上,要求每张表都只能表达一个意思.表的每个字段都应该与表的主键有依赖关系.**

员工表(emp1): 员工编号 员工姓名 部门名称 订单编号 不符合第二范式

员工表(emp2): 员工编号 员工姓名 部门名称 符合第二范式

### 第三范式:

**在第二范式的基础之上,其他的字段和主键直接只能有直接依赖关系.**

员工表(emp1): 员工编号 员工姓名 部门名称

部门表:部门编号 部门名称

第一个案例:不符合第三范式.因为用部门名称会产生数据的冗余.

员工表(emp2): 员工编号 员工姓名 部门编号

部门表:部门编号 部门名称

第二个案例:符合第三范式.