

## 4.2 存储过程 stored procedure

■ 25

■ 局部变量

■ 存储过程介绍

■ 创建和执行存储过程的语法

■ 不带参数存储过程

■ 带输入参数存储过程

■ 带输出参数存储过程

■ 20

■ 利用存储过程获取

“锦途网”综合数据

## 4.2.1 局部变量

- 局部变量（local variable）必须定义在存储程序中，如函数、存储过程、触发器以及事件中，而且局部变量的作用范围仅局限于存储程序中。如果脱离存储程序，局部变量将没有丝毫意义

### 语法

```
declare 局部变量 数据类型 ;
```

```
declare price decimal(8,2); declare address varchar(20);
```

## 4.2.1 局部变量

■局部变量主要应用于以下 3 种场合：

- 场合 1：局部变量定义在存储程序的 begin-end 语句块之间时，局部变量必须先进行 declare 命令定义，并且必须指定其数据类型。只有定义局部变量后，才可以使用 set 命令或 select 语句为其赋值。
- 场合 2：局部变量作为存储过程或函数的参数使用时，虽然不需要使用 declare 命令定义，但需要指定参数的数据类型。
- 场合 3：局部变量也可以用于存储程序的 SQL 语句中。数据检索时，如果 select 语句的结果集是单个值，则可以将 select 语句的返回结果赋予局部变量。局部变量也可以直接嵌入到 select 语句、insert 语句、update语句以及 delete 语句的表达式中

## 4.2.2 存储过程介绍

- 存储过程是一组为了完成特定功能的 SQL 语句集，经编译后存储在数据库中
- 用户通过指定存储过程的名字并给定参数（如果该存储过程带有参数）来调用执行它。
- 一个存储过程其实就是一个可编程的函数（函数有返回值，存储过程没有返回值），它在数据库中创建并保存，并由 SQL 语句和一些特殊的控制结构所组成。
- 当希望在不同的应用程序或平台上执行相同的功能，或者封装特定的功能时，使用存储过程是非常实用的解决之道

## 4.2.2 存储过程介绍

- 存储过程的优点主要包括以下 5 点
  - 增强了 SQL 语言的功能性和灵活性
  - 存储过程被创建后，可以在程序中被多次调用，而不必重新编写该存储过程的 SQL 语句
  - 能实现较快的执行速度
  - 能减少网络流量
  - 还可被作为一种安全机制来充分利用

## 4.2.3 创建和执行存储过程的语法

■ 创建存储过程的语法格式如下

### 语法

```
create procedure 存储过程名字 ([in|out|inout] 参数 1 数据类型 1,  
    [in|out|inout] 参数 2 数据类型 2,      .....)  
[no sql | reads sql data | modifies sql data]begin存储过程语句块 ;  
end;
```

## 4.2.3 创建和执行存储过程的语法

语法说明如下：

- 存储过程的参数是局部变量。
- in 代表输入参数（默认为 in 参数），表示该参数的值必须由调用程序指定。
- out 代表输出参数，表示经过存储过程的计算后，将 out 参数的计算结果返回给调用程序。
- inout 代表既是输入参数又是输出参数，表示该参数的值既可以由调用程序指定，又可以将该参数的计算结果返回给调用程序。

■ 执行存储过程的语法格式如下

语法

call 存储过程名 ( 参数列表 )

如果存储过程有参数，则需向存储过程传递 in 参数、out 参数或 inout 参数

## 4.2.4 不带参数存储过程

- 创建一个名为 `proc_product_info` 的存储过程，其将获取所有商品的标题、类型名、团购价、地区名和商店名，并按照类型和团购价升序显示

### 示例

```
begin
select title, categoryName, currentPrice, areaName, shopName from product p,
category c, Area a, Shop s where p.categoryID=c.categoryID and p.areaID=a.areaID and
p.shopID=s.shopID order by categoryName, currentPrice;
end
$$ delimiter ;
```



## 4.2.4 不带参数存储过程

- 在 Navicat For MySQL 单击“工具” “命令行界面”，出现 MySQL 命令行界面，在该界面输入上述存储过程的创建代码，按回车键即成功创建了存储过程

proc\_product\_info



```
mysql-1 - 命令行界面 x
文件(F) 编辑(E) 查看(V) 窗口(W)
[停止] [保存] [载入] [剪切] [复制] [粘贴] [清除] [自动换行]

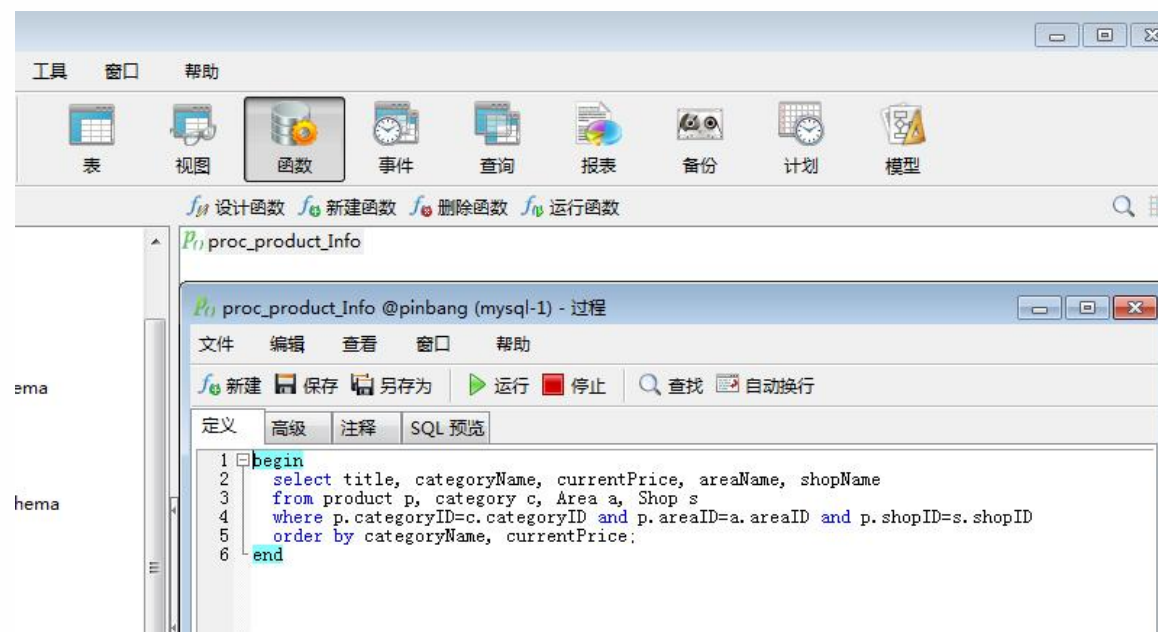
mysql> delimiter $$
create procedure proc_product_info()
reads sql data -- 读取数据
begin
    select title, categoryName, currentPrice, areaName, shopName from product p,
category c, Area a, Shop s where p.categoryID=c.categoryID and p.areaID=a.areaID
and p.shopID=s.shopID order by categoryName, currentPrice;
end
$$
delimiter ;

Query OK, 0 rows affected

mysql>
```

## 4.2.4 不带参数存储过程

■在 Navicat For MySQL 的“函数”处可见 proc\_product\_info，表明 proc\_product\_info 已创建成功。单击需要操作的存储过程，如 proc\_product\_info，点击“设计函数”即可对该存储过程的代码进行编辑

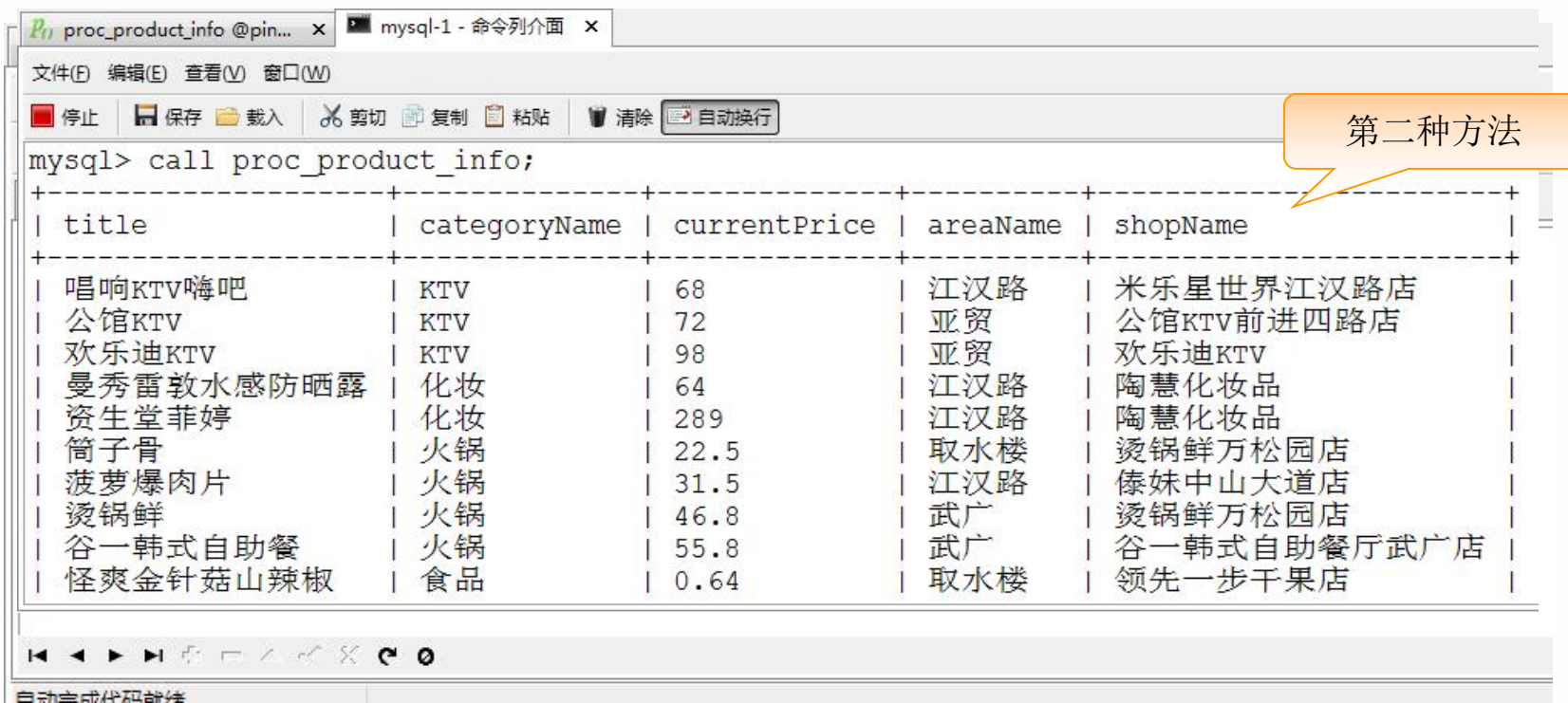


## 4.2.4 不带参数存储过程

■ 运行存储过程的两种方式如下

■ 在存储过程的编辑窗口，点击“运行”

■ 在 MySQL 命令行输入“call 存储过程名”，本例即为“call proc\_product\_info;”



第二种方法

```
mysql> call proc_product_info;
```

title	categoryName	currentPrice	areaName	shopName
唱响KTV嗨吧	KTV	68	江汉路	米乐星世界江汉路店
公馆KTV	KTV	72	亚贸	公馆KTV前进四路店
欢乐迪KTV	KTV	98	亚贸	欢乐迪KTV
曼秀雷敦水感防晒露	化妆	64	江汉路	陶慧化妆品
资生堂菲婷	化妆	289	江汉路	陶慧化妆品
筒子骨	火锅	22.5	取水楼	烫锅鲜万松园店
菠萝爆肉片	火锅	31.5	江汉路	傣妹中山大道店
烫锅鲜	火锅	46.8	武广	烫锅鲜万松园店
谷一韩式自助餐	火锅	55.8	武广	谷一韩式自助餐厅武广店
怪爽金针菇山辣椒	食品	0.64	取水楼	领先一步干果店

## 4.2.4 不带参数存储过程

- 创建一个名为 `proc_ProductStatistics` 的存储过程，将获取不同类型的商品服务的个数和平均团购价

### 示例

```
begin
select categoryName 商品类型名 , count(p.productID) 商品数量 , avg(currentPrice)
平均团购价 from product p, category c where p.categoryID=c.categoryID
group by categoryName order by 平均团购价 ;
end
$$
delimiter ;
```

## 4.2.5 带输入参数存储过程

- 创建一个名为 `proc_OrdersGivenCustomer` 的存储过程，将获取指定客户在指定日期之后的订购信息，要求输出客户姓名、订单编号、下单日期、商品标题和团购价，按照订单编号和团购价升序显示

### 示例

```
delimiter $$  
  
create procedure proc_OrdersGivenCustomer(  
in _customerName varchar(20),  
in _ordersDate date)
```

需要两个输入参数：一个用于接收客户信息（本题为客户姓名）；另一个用于接收指定日期

## 4.2.5 带输入参数存储过程

```
reads sql data
begin
select customerName 客户姓名 , o.ordersID 订单号 , ordersDate 下单日期 ,
title 商品标题 , currentPrice 团购价
from customer c, orders o, ordersDetail od, product p
where c.customerID=o.customerID and o.ordersID=od.ordersID
and od.productID=p.productID and c.customerName=_customerName
and ordersDate>_ordersDate order by o.ordersID, currentPrice;
end
$$
delimiter ;
```

## 4.2.5 带输入参数存储过程

- 执行该存储过程，用于获取姓名为“雷亚波”的客户于 2019 年 3 月 31 日之后的订购信息，执行存储过程 `proc_OrdersGivenCustomer`

```
set @customerName= ' 雷亚波 ' ;  
set @ordersDate= '2019-3-31 ' ;  
call proc_OrdersGivenCustomer(@customerName, @ordersDate);
```

### 注意

执行带参数的存储过程时，传入值的类型、个数和顺序都需要与存储过程中定义的参数逐一对应。

## 4.2.6 带输出参数存储过程

- 如果需要存储过程返回一个或多个值，则可通过使用输出参数来实现。输出参数必须在创建存储过程时，使用out 关键字进行声明。
- 创建一个名为 `proc_MaxPriceGivenCategory` 的存储过程，其将获取指定类型的商品的最高团购价

### 示例

```
delimiter $$  
  
create procedure proc_MaxPriceGivenCategory(  
    _categoryName varchar(20),  
    out _maxPrice decimal    )
```

需定义两个参数：一个为输入参数，用于接收指定商品类型信息（本题为类型名）；另一个为输出参数，用于输出指定类型的商品的最高团购价



## 4.2.6 带输出参数存储过程

```
reads sql data
begin
select max(currentPrice) into _maxPrice from product p, category c
where p.categoryID=c.categoryID and categoryName=_categoryName;
end
$$
delimiter ;
```

执行该存储过程

```
set @categoryName=' 火锅 ' ;
call proc_MaxPriceGivenCategory(@categoryName, @maxPrice);
select concat(@categoryName,' 火锅类商品的最高团购价是 ',@maxPrice,' 元 ') 显示结果 ;
```

## 4.2.7 学生实践练习

- 需求说明（1）的解决思路

- 创建存储过程 `proc_LineDetail` 仅含有一个输入参数，用于接收线路名

- 执行存储过程

- 需求说明（2）的解决思路

- 存储过程 `proc_proc_LineNumsRate` 含有两个参数：一个是输入参数，用于接收线路类型名称；另一个是输出参数，用于输出指定类型的线路数与全部线路数之比

- 执行存储过程