

4.1 用户自定义变量

■ 25

■ 设计“优乐网”数据库

■ 数据定义语言

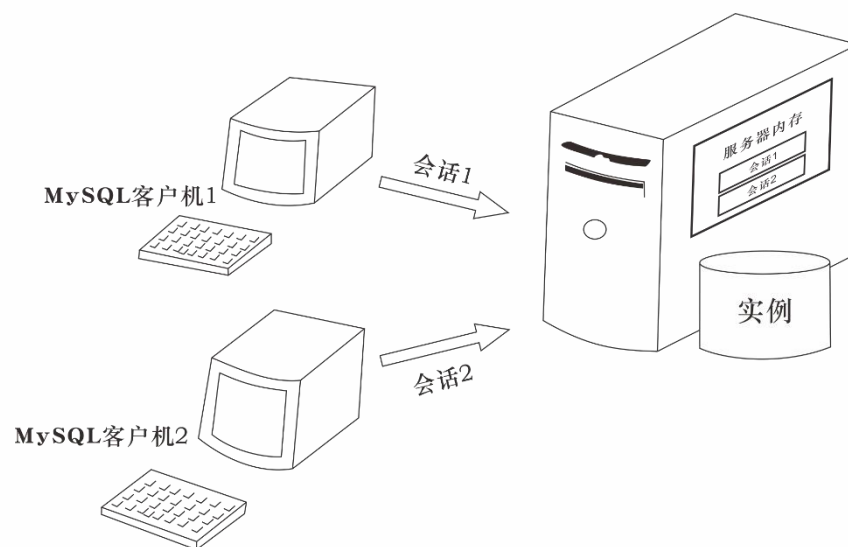
■ 数据操作语言

■ 20

■ 使用用户自定义变量定义
“锦途网”数据

4.1.1 用户会话变量 User sessions variable

- 用户自定义变量用于存储 MySQL 存储程序运行期间所产生的临时变量，它分为用户会话变量和局部变量
- 用户会话变量（User Session Variable）是在每次 MySQL 客户与 MySQL 服务器建立一个新的连接后，由MySQL 客户进行定义的一种变量，该变量与“当前会话”有密切关系。



4.1.1 用户会话变量

- MySQL 用户会话变量以一个“@”开头，并且大小写不敏感。一般情况下，用户会话变量的定义和赋值会同时进行。
- 使用 `set` 命令和 `select` 语句，可以对用户会话变量进行定义和赋值
- 使用 `set` 命令定义用户会话变量的语法格式如下

`user_variable1`、
`user_variable2` 为用户
会话变量名；
`expression1`、
`expression2` 可以是常
量、变量和表达式

```
set @user_variable1=expression1[,@user_variable2=expression2, ...]
```

4.1.1 用户会话变量

- 使用 `set` 命令创建 MySQL 用户会话变量 `@user_name` 以及 `@age`，并为其赋值，然后使用 `select` 语句输出上述变量的值

```
set @user_name= ' 张三 ' ;  
set @age=18;  
select @user_name, @age;  
set @age=@age+1;  
select @user_name, @age;
```

4.1.1 用户会话变量

■使用 `select` 语句定义用户会话变量的语法格式如下

语法

第一种：“`select @user_variable1:=expression1[,@user_variable2:=expression2, ...]`”

第二种：“`select expression1 into @user_variable1, expression2 into @user_variable2, ...`”

注意

第一种语法格式与第二种语法格式的区别在于：第一种语法格式中的 `select` 语句会产生结果集，第二种语法格式中的 `select` 语句仅用于会话变量的定义及赋值，但不会产生结果集。

4.1.1 用户会话变量

示例

- 使用select 语句创建 MySQL 用户会话变量 @user_name，并为其赋值，然后使用 select 语句输出该变量的值

```
select @user_name:='zhangsan';  
select @user_name;  
select 'zhangsan' into  
@user_name;  
select @user_name;
```

4.1.2 用户会话变量赋值 User sessions variable assignment

- 检索数据时，如果 select 语句的结果集是单个值，可以将 select 语句的返回结果赋予用户会话变量。
- 统计“零聚网”商品和服务的数量，并将数量赋给用户会话变量 @productNum，并输出该变量值

示例

```
select @productNum:=count(*) from product （简化形式 1）  
select count(*) into @productNum from product （简化形式 2）  
select count(*) from product into @productNum （简化形式 3）
```

方法 3：去掉子查询
的简化形式（最常见）

4.1.3 重置命令结束标记 Reset ‘end mark’ command

- begin-end 语句块中通常存在多条 MySQL 表达式，每条 MySQL 表达式都使用 “;” 作为结束标记。
- 在 MySQL 客户机上输入 MySQL 命令或 SQL 语句时，默认情况下 MySQL 客户机也是使用 “;” 作为 MySQL 命令的结束标记。
- 由于 begin-end 语句块中的多条 MySQL 表达式密不可分，为了避免这些 MySQL 表达式被拆开，需要重置 MySQL 客户机的命令结束标记，亦称命令分隔符 (delimiter)

4.1.3 重置命令结束标记

- 通过重置命令结束标记，显示商品和服务的大类类型信息，大类的p_categoryID为空

示例

命令结束标记“临时地”设置为“\$\$”

```
delimiter $$
```

```
select * from category where p_categoryID is null $$
```

使用“\$\$”作为 SQL 语句的结束标记

```
delimiter ;
```

命令结束标记恢复“原状”

重新使用“;”作为结束标记

```
select * from category where p_categoryID is null;
```

4.1.4 学生实践练习

- (1) 统计“锦途网”旅游线路平均价格，并采用尽可能多的方式将该价格赋给用户会话变量 `@avg_short_price`，并输出该变量。
- (2) 在 MySQL 客户端重置命令行结束符为“\$\$”，并依次定义一个变量 `@avg_short_price` 为 `decimal(8,2)`，然后将旅游线路平均价格赋给变量 `@avg_short_price`，并输出该变量，最后将命令行结束符恢复为“;”

4.1.4 学生实践练习

■需求说明（1）的解决思路

- 获取“锦途网”旅游线路平均价格的 SQL 语句

- 将以上平均价格赋给用户会话变量 @avg_short_price: set+ 子查询、select+ 子查询或者去掉子查询的简化形式的方法

- 使用 select 语句输出变量 @avg_short_price

■需求说明（2）的解决思路

- 可以使用delimiter关键字