

**课程设计报告**

( 2022- 2023年度第1学期)

实验名称： 数据库应用课程设计

题 目： 网上购物系统

院 系： 控制与计算机工程学院

班 级： 信安2002班

组 号： 9

指导教师： 王素琴

设计周数： 1 周

成 绩：

日期：2023年1月14日

1. **成员及分工**

**学号：**120201080418**姓名：赵国利** **主要工作 数据库设计与实施、前端功能设计、报告撰写**

**学号：**120201080614**姓名：马鑫 主要工作 前端功能实现、数据库测试、数据库功能完善**

**学号：**120201080518**姓名：平措朗杰** **主要工作 数据库需求分析、ppt制作、数据库后期运维**

1. **课程设计的目的与要求**

**目的**

1. 学习和实践在分析和设计计算机应用系统所需要的知识，包括面向对象的系统分析与设计，对数据库做进一步的了解，掌握相关知识；
2. 进一步加强对数据库运用能力和熟练掌握数据库中的重要知识，了解如何从数据库中读写有关数据；
3. 培养分析问题、解决问题的能力。

**要求**

1. 完成数据库系统的安装与设置。
2. 根据具体的课题完成需求分析。
3. 完成数据库应用系统的逻辑设计。
4. 创建数据库、数据表。
5. 完成设计报告。

**二、设计正文**

# 1.需求分析

1.1 调查用户需求

随着互联网技术的发展，电商市场已经进行的如火如茶，网上购物已经是趋势网上很多销售商不方便对市场的需求进行调查.对自己所卖的商品不能及时地了解相关信息.获利没有达到最大限度:当购物者需要买自己所需的商品时，需要浪费很多的财力和物力。在中国电商行业飞速发展的今天。互联网成为人们快速获取、发布和传递信息的重要柒道。网上购物系统的开发可以提供更多的便利.网上购物弥补了传统购物的一些局限性。而且这对于管理人员来说，利用购物系统能轻松地管理大量的商品，而且可以通过系统查看各种商品的销售状况，通过各种数据能够及时地知道那种商品正在热实。而那些商品属于冷门地。从而在运营上进行调整，更好地管理好商店。从而达到更多的销售业绩。

我们设计的网上购物系统的数据库开发存储，其可以存储销售商需要卖出的商品信息用户的所有信息.方便用户根据自己的喜好，浏览自己喜欢的东西.若要想购买商品，通过注册成为会员进行购买，登陆后就可以购物了。商店的各种商品都进行了详细地分类，可以轻松地找到想要的商品，同时查找系统能很容易地找到相关的商品。同时买到自己喜欢的东西后就可以去结算.通过填写自己的相关信息，以便下订单，从而完成购物的流程。  
 现在准备建立网上购物系统。要把销售商的商品、会员的信息、会员购买商品的记录等基本信息进行管理。确保有足够的安全性，以便网上购物系统可以进行推广，让更多的销售商和购物者认为网上购物系统非常实惠便利的，是一个对非常有用的购物系统。

本系统的最终用户为注册成为会员的购物者，购物系统内部的管理人员。根据我们日常生活中的经验，结合对网上购物系统的参考和用户购物调查，得出用户的下列实际要求：

**1.1.1基本功能需求**

商铺店主希望借助网上系统销售商品，用户登录后，才可以实现购物功能，并且可以查看订单情况。用户购买的商品都放入购物车中，为了方便用户搜索自己需要的商品，还要能够进行商品的分类和查找。系统主要信息模块如下：

* 会员注册信息

用户注册成为会员时，每位会员都有唯一的会员编号，并记录登记的会员姓名、密码、电话、地址等信息。会员编号和密码记录在数据库中用于会员登陆，会员电话用于与会员进行联系及售后处理，会员地址用于会员下单后邮寄商品。

* 商品具体信息  
   每个商品都有唯一的商品编号，不同种类的商品分为不同的商品组并赋予商品组编号，上架商品时还会记录商品名称、标注商品价格，管理员可以对商品进行简介编写方便用户了解商品更多信息，也可以对商品信息进行维护。用户可查看商品信息同时商品还有图片示例展示可供用户挑选心仪商品。此外，用户还可以进行商品查找，查找特定商品或该类所有商品信息，平台也会展示出销售量前六位的商品信息推荐给用户。
* 购物车信息  
   每个用户可以将商品添加在购物车内并可在购物车中添加或删除商品。每个购物车都有一个购物车编号，数据库后台还会记录购物车编号对应的用户编号。购物车中记录所购商品编号和商品购买数量。
* 订单信息  
   用户对商品进行购买下订单后会生成唯一订单编号，并且记录订单日期、最后总价、商品编号和会员编号等信息，方便后续进行用户、订单、商品的匹配。用户可以维护订单信息，并查找到自己的所有订单信息。
* 管理员信息

数据库内存放系统管理员的基本信息，包括管理员编号、管理员姓名、密码、联系电话等，用于管理员登陆系统进行维护和与管理员进行联系沟通。

* 商品组信息  
   同种类的商品存放在同一个商品组并拥有唯一的商品组编号，同时还记录商品组名称、管理员编号、以及商品简介描述。每个商品组由特定管理员进行管理。
* 管理员—权限信息  
   由于不同级别的管理人员拥有不同的权限，所以需要管理员—权限信息用于记录不同管理员的不同权限，可记录管理员编号和权限编号，并且由二者可唯一确定管理员—权限信息。
* 权限信息  
   将所有的权限信息记录在一起用于后续统计及日常工作的开展，其中每个权限有相应的权限编号，权限名称和权限描述，每个权限信息可由权限编号唯一确定。

**1.1.2用户对系统的要求**

* **网上购物系统管理员**

**a.信息要求**

网上购物系统管理员能查询上面提到的上架商品的所有相关信息，包括商品编号、商品组编号、名称、价格、简介、图片等。

不同级别的管理人员能够管理查看会员信息、订单信息、购物车信息等，同时还可以查看权限信息和相应的管理员—权限信息。

**b.处理要求**

当商品信息发生变化时，系统管理员能对其进行修改。比如，某些商品下架，它们在本系统中相应的信息就应删去；或商品图片需要更新，它们的信息中图片要重新上传进行相应的修改等等。

当商品组信息发生变化时，应由相应级别的管理员对该商品组的信息进行更改并确定该商品组新的管理员编号和描述。

当会员信息需要进行处理时，可由后台管理员进行处理。

管理员可对用户订单、购物车等信息进行处理。

当管理员权限发生变化时，由相应级别的管理员进行权限信息的更新并对管理员—权限信息进行更改。

**c．安全性与完整性要求**

安全性要求

* + 系统应设置访问用户的标识以鉴别是否是合法用户，并要求合法用户设置其密码，保证用户身份不被盗用；
  + 系统应对不同的数据设置不同的访问级别，限制访问用户可查询和处理数据的类别和内容；
  + 系统应对不同用户设置不同的权限，区分不同的用户，如区分普通用户和不同级别的管理员。

完整性要求

* + 各种信息记录的完整性，信息记录内容不能为空；
  + 各种数据间相互的联系的正确性；
  + 相同的数据在不同记录中的一致性。
* **网上购物用户**

**a.信息要求**

本系统用户能查询所售商品的所有信息，能查询自己购物车相应信息及订单信息，用户可以查找特定商品或商品组信息同时可以查看销量最多的前六名商品。

**b.处理要求**

本系统用户可以对自己的购物车进行管理，可以随意添加或删除商品并查看商品数量。

用户可以自行维护订单信息，并查找自己的所有订单信息。

用户可在购物车内或商城中直接下订单并完成付款。

1.2 系统功能的设计和划分

根据如上得到的用户需求，我们将本系统按照所完成的功能分成以下几部分：

**第一部分：用户管理部分**

**第二部分：管理员管理部分**

各部分完成的功能如下：

**1.2.1用户管理部分**

* + 处理用户注册
  + 处理用户登录
  + 用户可以查询商品信息。
  + 用户可以查询购物车信息。
  + 用户可以提交与查询订单信息。
  + 用户可以增加与删除用户自己的购物车物品。
  + 用户可以修改用户自己密码。
  + 用户可以查找特定商品信息及商品组信息
  + 用户可以查看销量最高的前六名商品

**1.2.2管理员管理部分**

* + 处理管理员登录
  + 管理员可以查询商品信息。
  + 管理员可以发布与更新商品信息。
  + 管理员可以发布与更新商品组信息。
  + 管理员可以插入、修改与查询权限信息。
  + 管理员可以插入、修改与查询管理员—权限信息。
  + 管理员可以查询用户订单信息。
  + 管理员可以查询管理会员信息
  + 管理员可由对应级别管理员查询管理管理员信息
  + 管理员可以修改管理员密码。

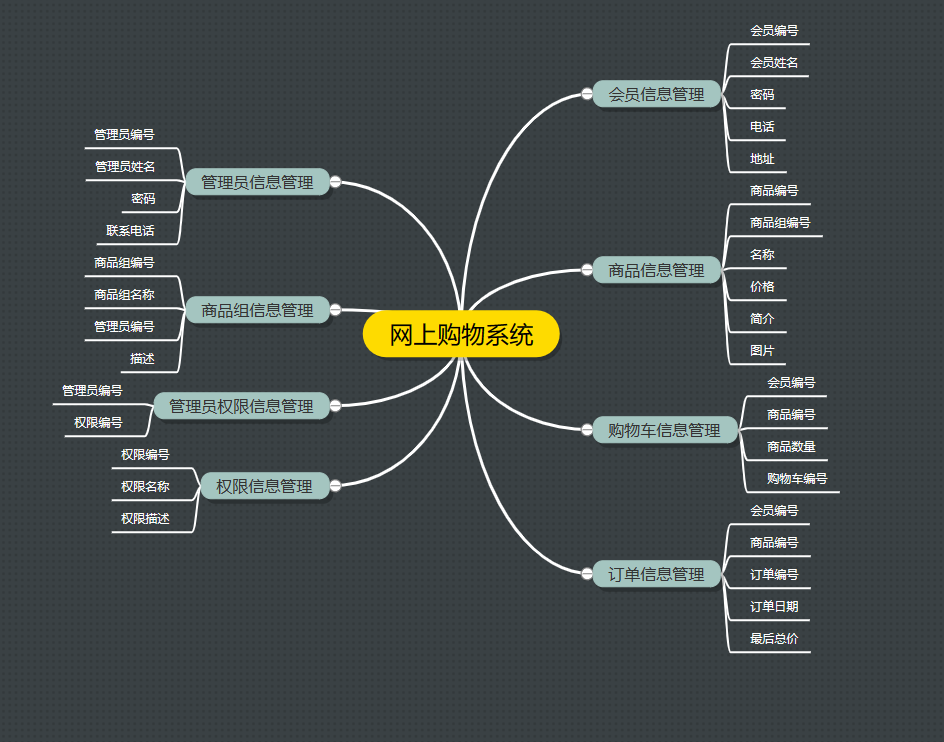


图1.系统功能图

1.3 数据流图

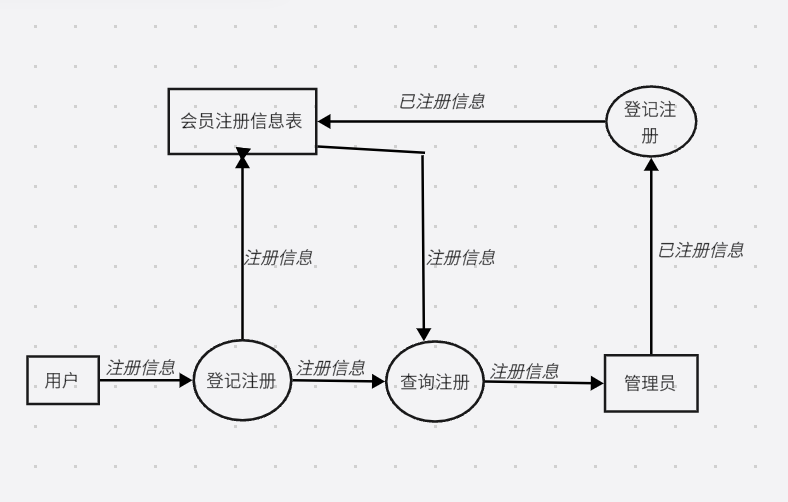


图2.会员注册信息分数据流图

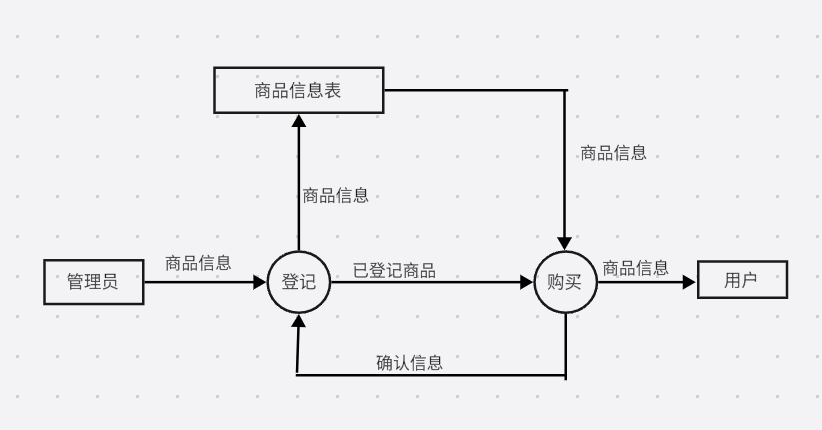


图3.商品/商品组信息分数据流图

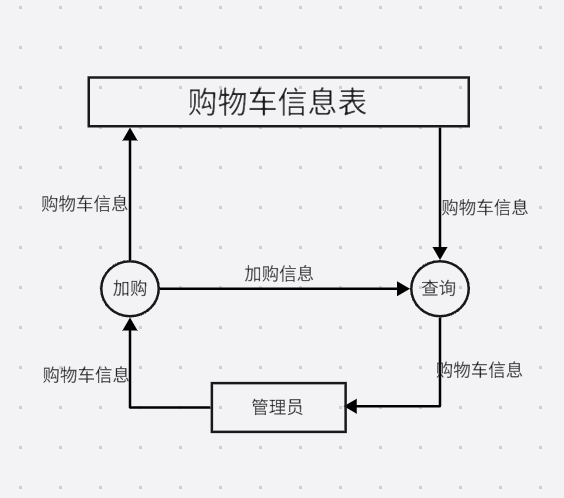


图4.购物车信息分数据流图

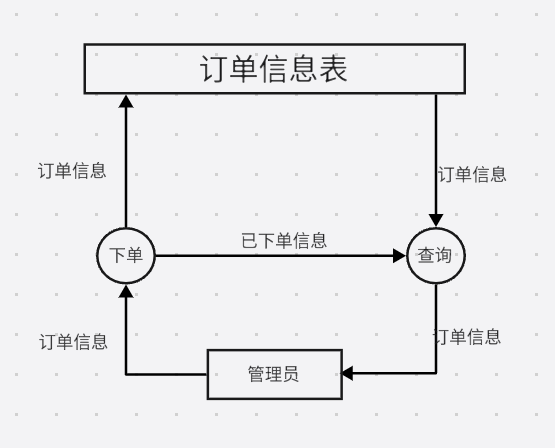


图5.订单信息分数据流图

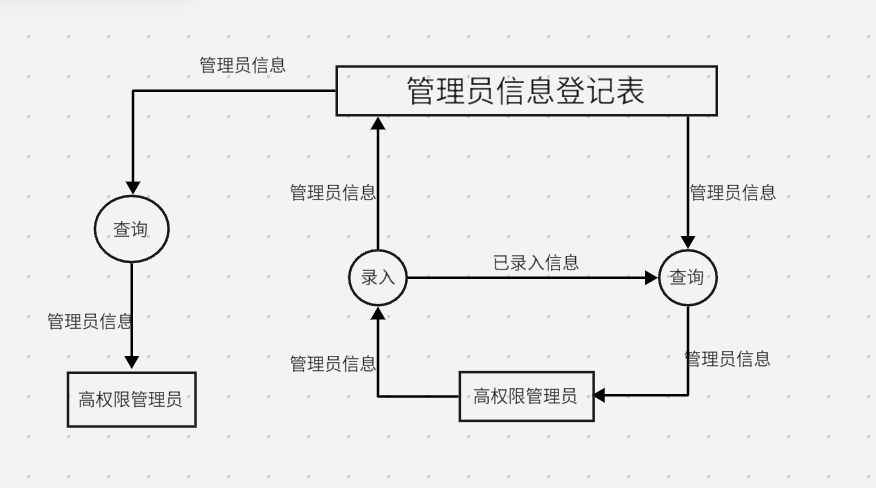


图6.管理员信息分数据流图

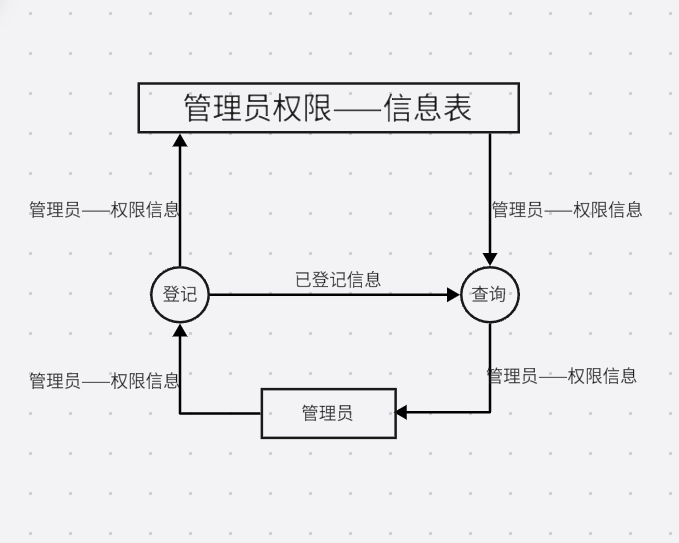


图7.管理员—权限信息分数据流图

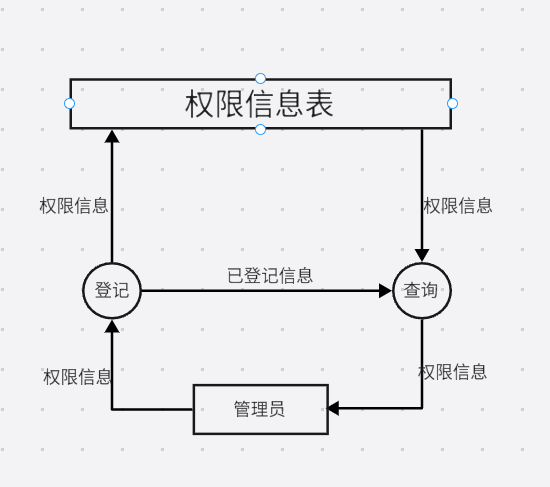


图8.权限信息分数据流图

# 2.概念设计

根据需求分析说明书设计ER图，设计完成的ER图为：

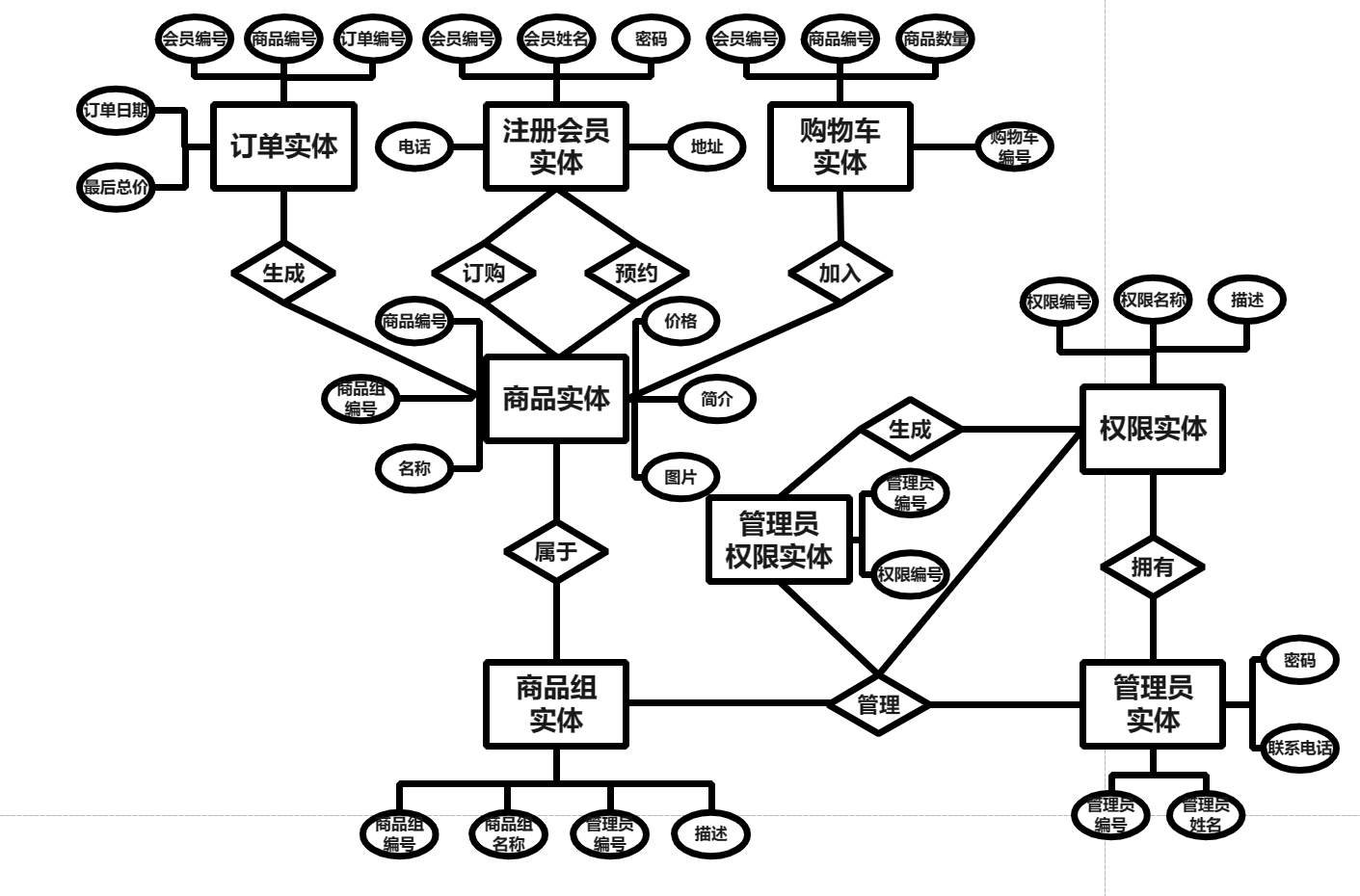


图9. 总ER图

# 3.逻辑设计

3.1 关系表设计

将ER图转换为关系模式、要求设计出数据库所有关系表

表1. 会员注册信息表（huiyuan）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 会员编号 | Hno | varchar | 10 |  | 主键，非空 |
| 姓名 | Hname | varchar | 20 |  | 非空 |
| 密码 | Hpassword | varchar | 20 |  | 非空 |
| 电话 | Htel | varchar | 20 |  | 非空 |
| 地址 | Hadd | varchar | 40 |  | 非空 |

表2.商品具体信息表（shangpin）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 商品编号 | Sno | varchar | 10 |  | 主键,非空 |
| 商品组编号 | SZno | varchar | 10 |  | 非空，外键为shangpinzu中的SZno |
| 名称 | Sname | varchar | 20 |  | 非空 |
| 价格 | Sprice | float |  |  | 非空 |
| 简介 | Sintro | varchar | 20 |  |  |

表3.购物车信息表（gouwuche）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 购物车编号 | Gno | int |  |  | 主键,非空 |
| 会员编号 | Hno | varchar | 10 |  | 非空，外键为huiyuan中的Hno |
| 商品编号 | Sno | varchar | 10 |  | 非空，外键为shangpin中的Sno |
| 商品数量 | Snum | int |  |  | 非空 |

表4.订单信息表（dingdan）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 订单编号 | Did | int |  |  | 主键，非空 |
| 会员编号 | Hno | varchar | 10 |  | 非空，外键为huiyuan中的Hno |
| 商品编号 | Sno | varchar | 10 |  | 非空，外键为shangpin中的Sno |
| 订单日期 | Ddate | datetime |  |  | 非空 |
| 最后总价 | Dtotal | float |  |  | 非空 |
| 物品编号 | Object\_id | varchar | 500 |  |  |

表5.管理员信息表（gly）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 管理员编号 | GLno | varchar | 10 |  | 主键,非空 |
| 管理员姓名 | GLname | varchar | 20 |  | 非空 |
| 密码 | GLpassword | varchar | 20 |  | 非空 |
| 联系电话 | GLtel | varchar | 20 |  | 非空 |

表6.商品组信息表（shangpinzu）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 商品组编号 | SZno | varchar | 10 |  | 主键,非空 |
| 商品组名称 | SZname | varchar | 20 |  | 非空 |
| 管理员编号 | GLno | varchar | 10 |  | 非空，外键为gly中的GLno |
| 描述 | SZintro | varchar | 20 |  |  |

表7.管理员权限信息表（glyquanxian）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 | 完整性约束 |
| 管理员编号 | GLno | varchar | 10 |  | 主键，非空，外键为gly中的GLno |
| 权限编号 | Qno | varchar | 10 |  | 非空,外键为quanxian中的Qno |

表8.权限信息表（quanxian）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 类型 | 长度 | 备注 | 完整性约束 |
| 权限编号 | Qno | varchar | 10 |  | 主键，非空 |
| 权限名称 | Qname | varchar | 20 |  | 非空 |
| 描述 | Qintro | varchar | 20 |  | 非空 |

表9.商品图片信息表（quanxian）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 字段名 | 类型 | 长度 | 备注 | 完整性约束 |
| 商品编号 | Sno | varchar | 10 |  | 非空 |
| 图片路径 | picture\_path | varchar | 60 |  |  |

**以上关系模式均为BCNF。**

3.2视图设计

为了方便程序查询，建立了如下**用户视图：**

商品信息视图（商品编号，商品组编号，名称，价格，简介，描述）

CREATE VIEW shangpinView

AS

SELECT Sno,shangpin.SZno,Sname,Sprice,Sintro

FROM shangpin,shangpinzu

WHERE shangpin.SZno=shangpinzu.SZno

订单信息视图（订单编号，会员编号，商品编号，最后总价，商品价格，下单日期）

CREATE VIEW dingdanView

AS

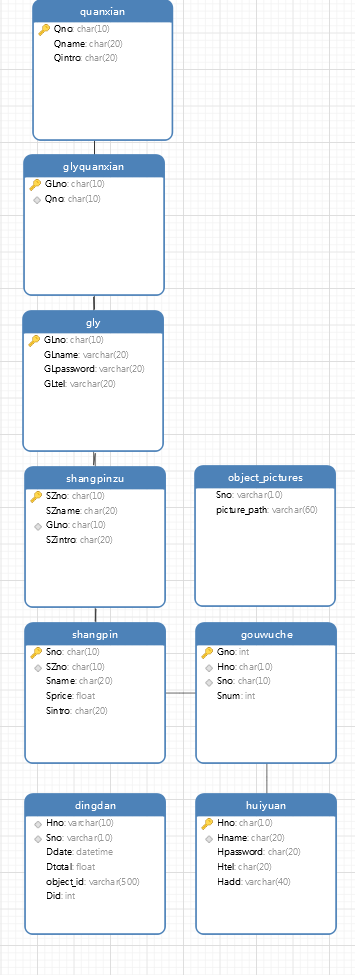
SELECT Dno,Hno,dingdan.Sno,Dtotal,Sprice,Ddate

FROM dingdan,shangpin

WHERE dingdan.Sno=shangpin.Sno

# 4.物理设计

**4.1全局数据表的关联图**

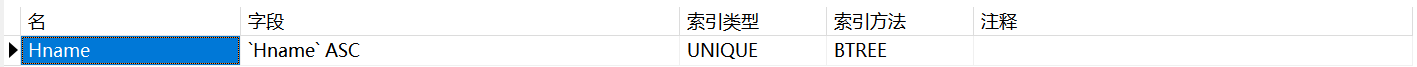


**4.2各表索引**



屏幕截图 2023-01-14 195400



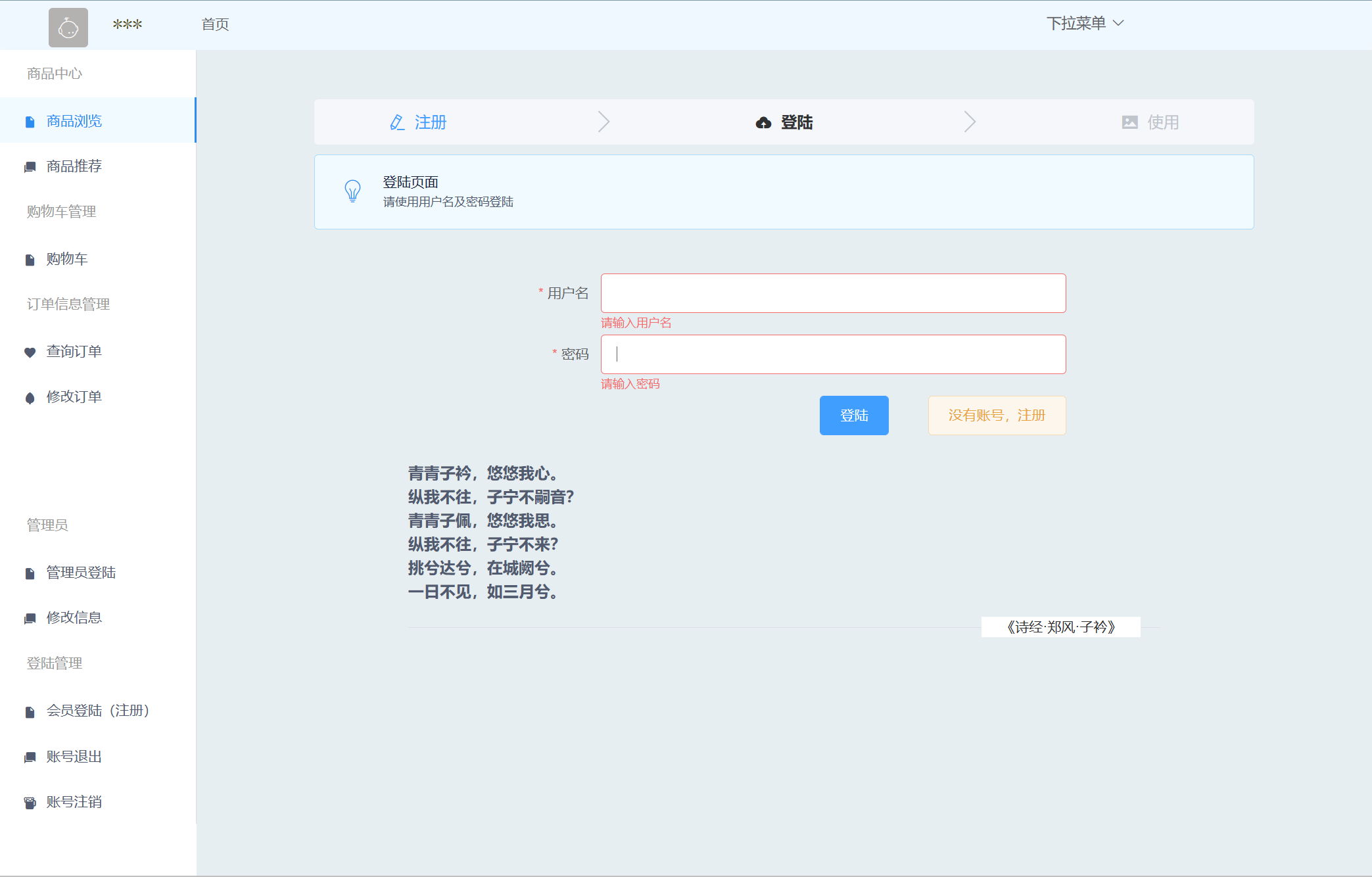


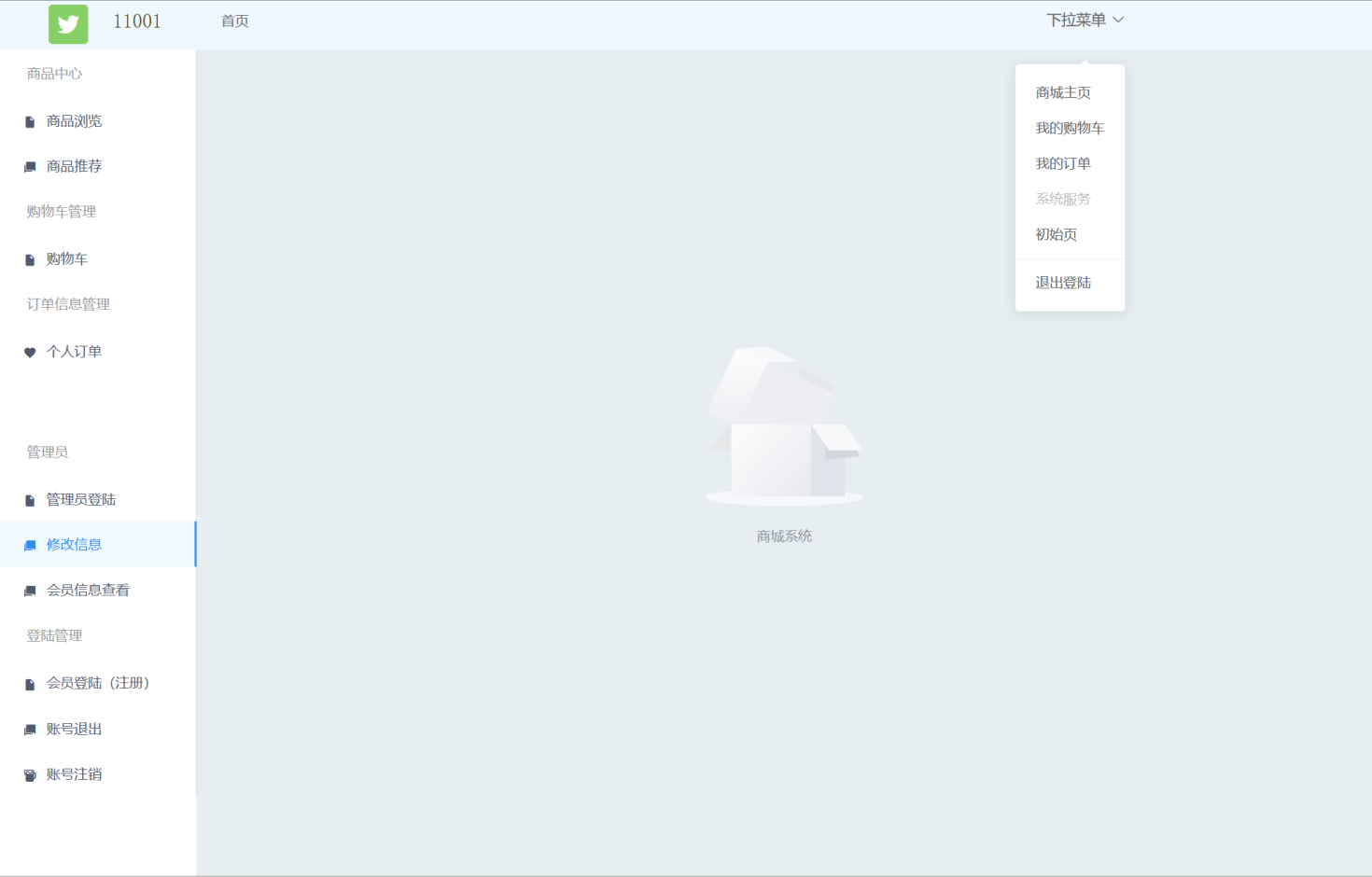
屏幕截图 2023-01-14 195458

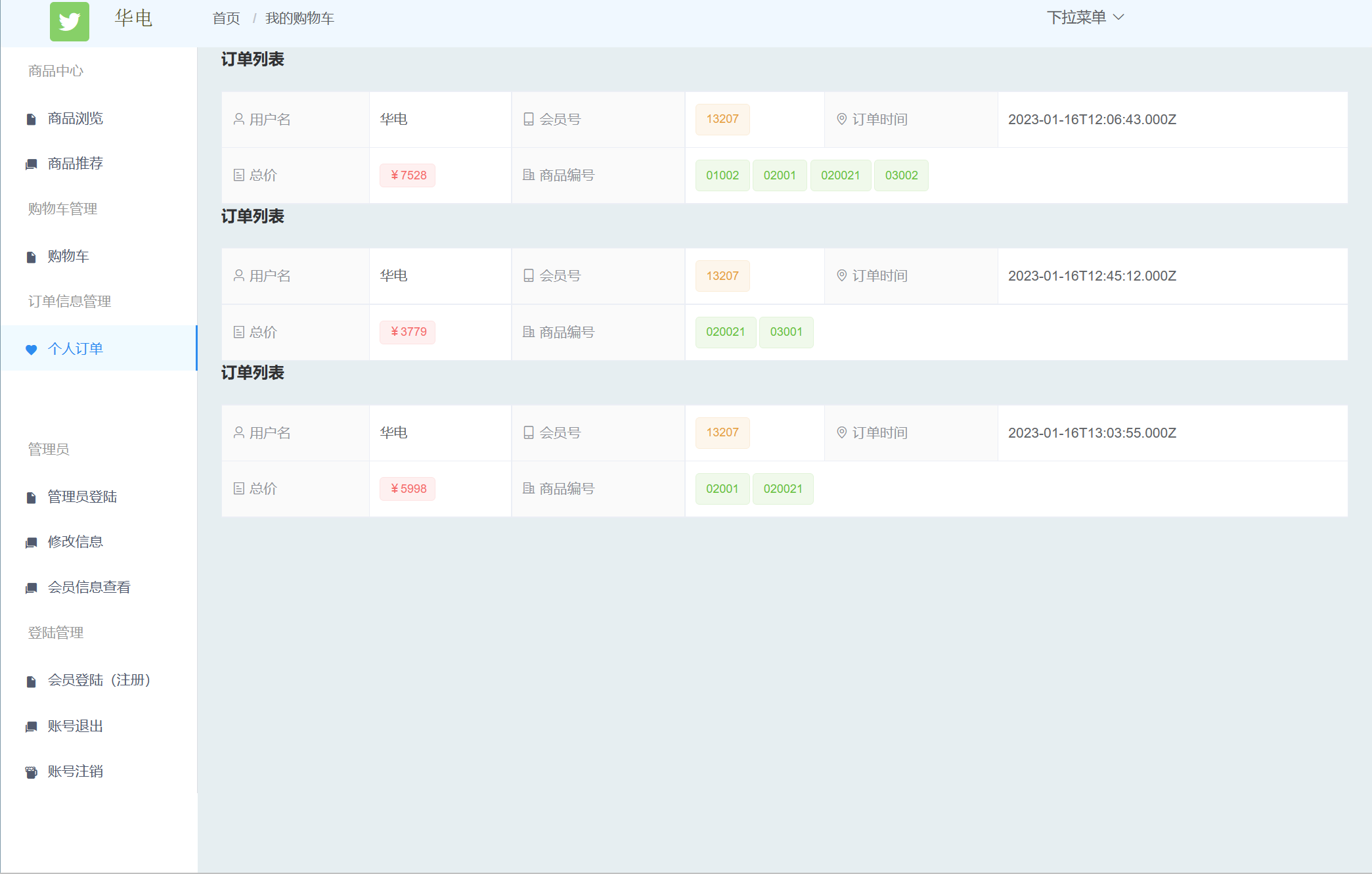


# 5.系统界面及代码实现

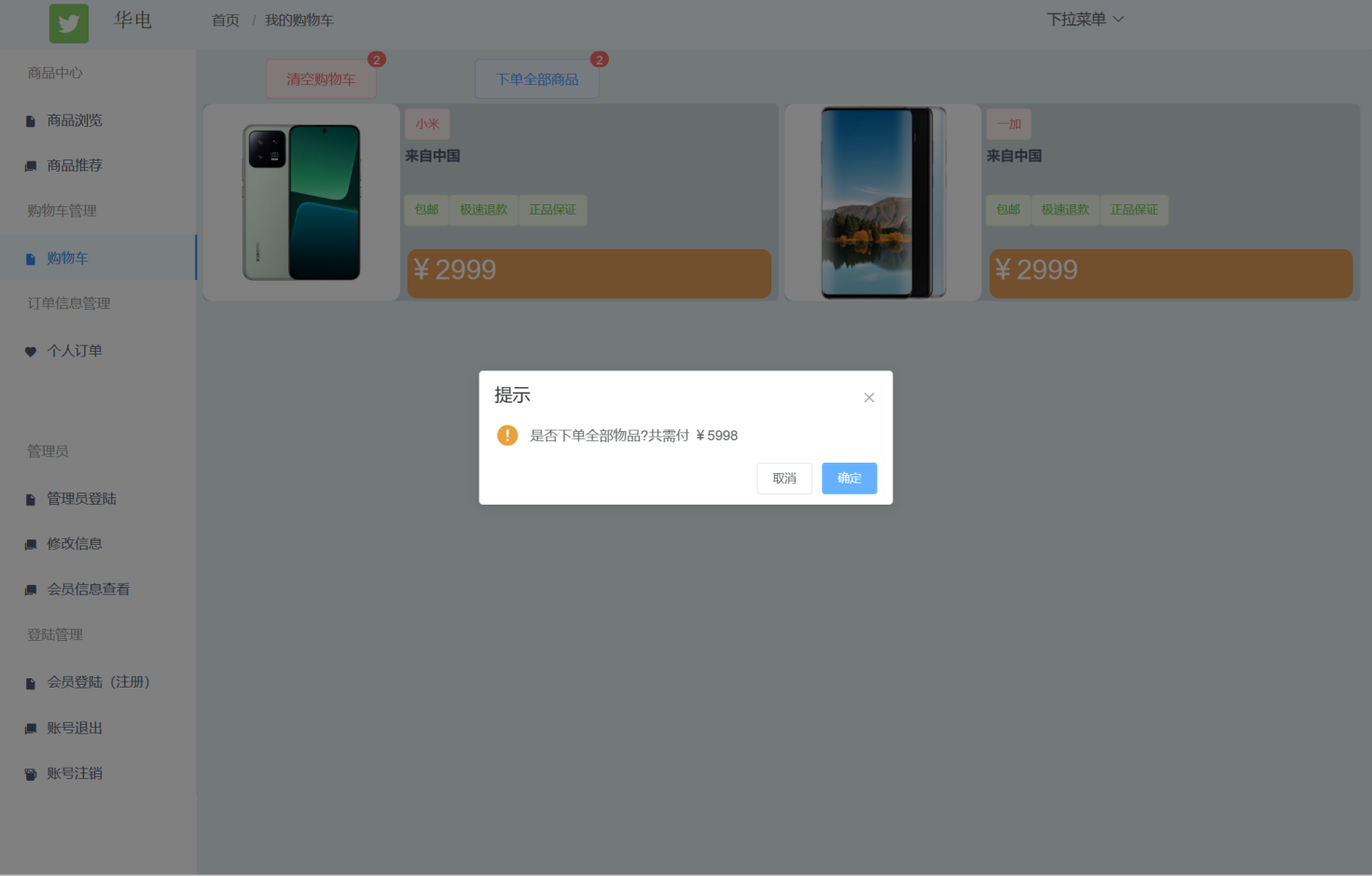
**5.1前端设计及代码实现**

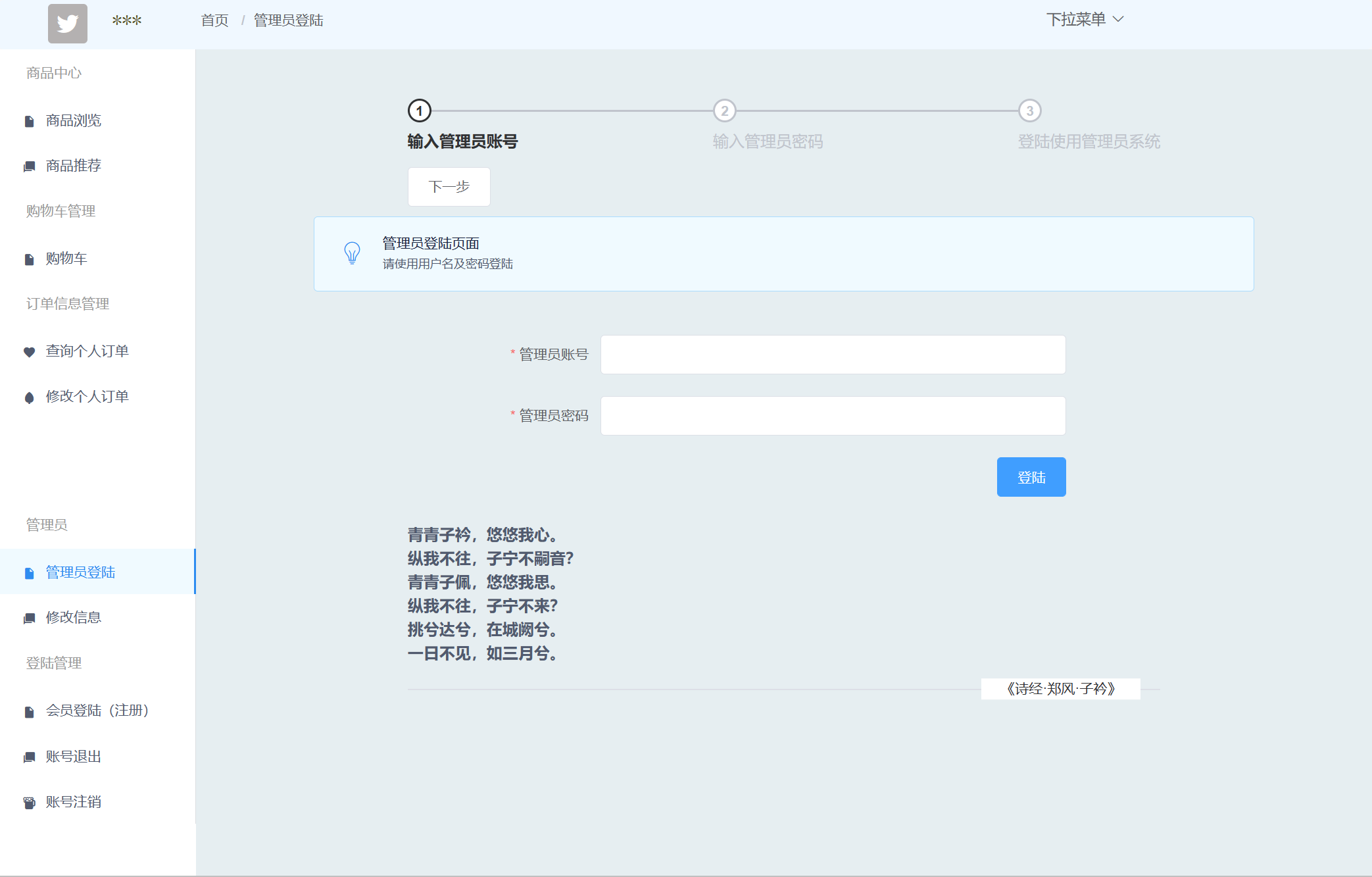
****

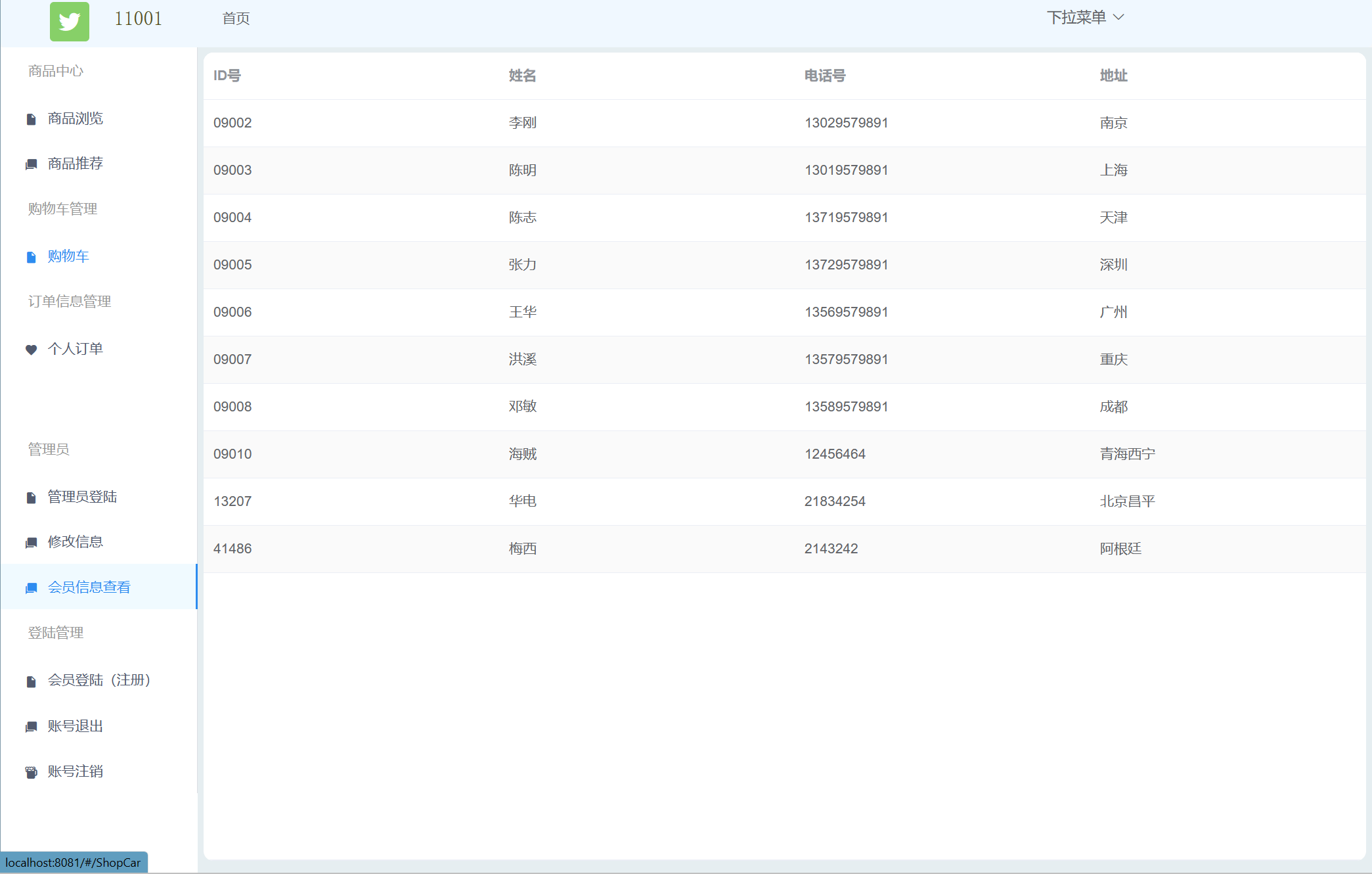
****

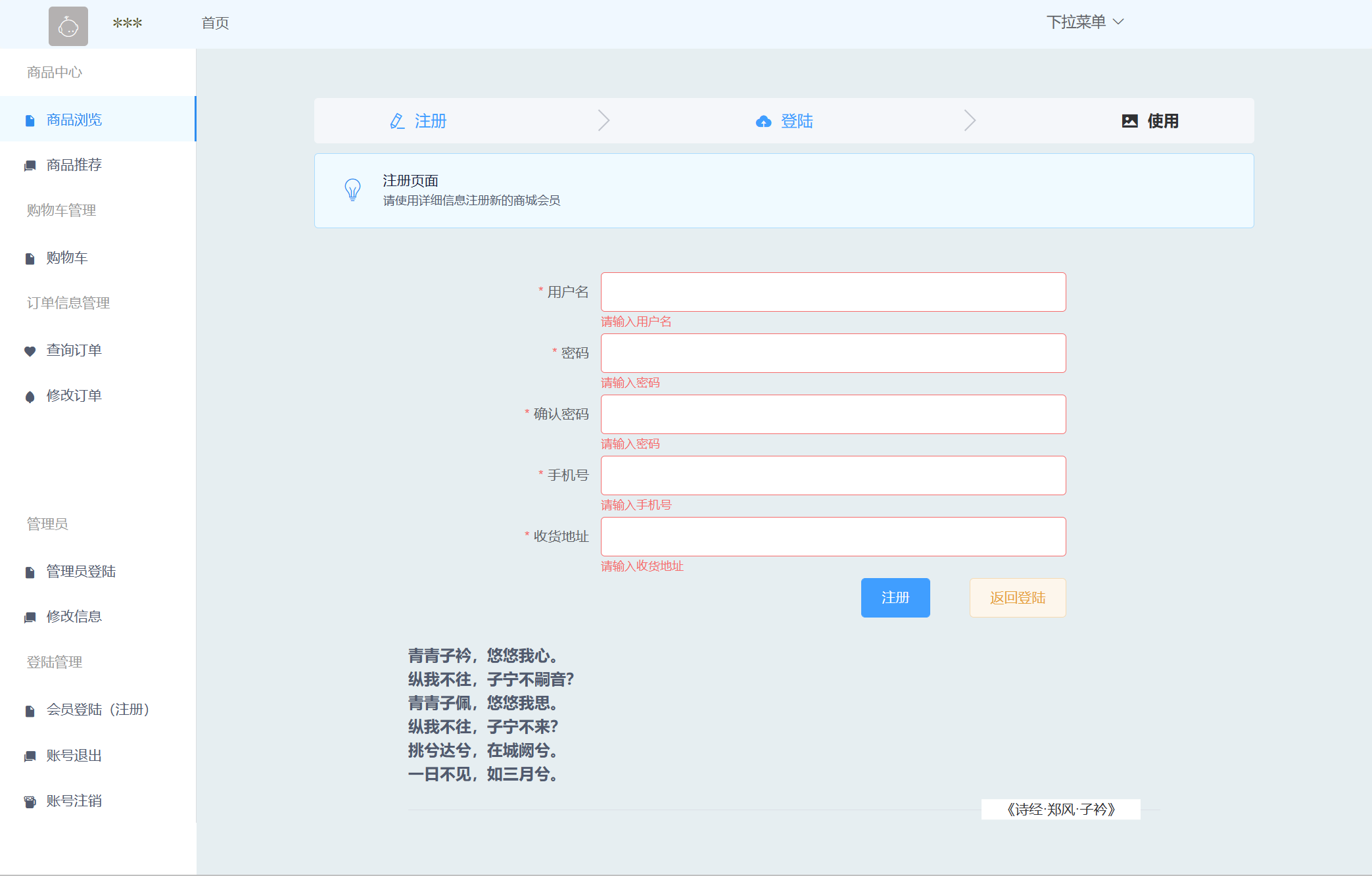
****

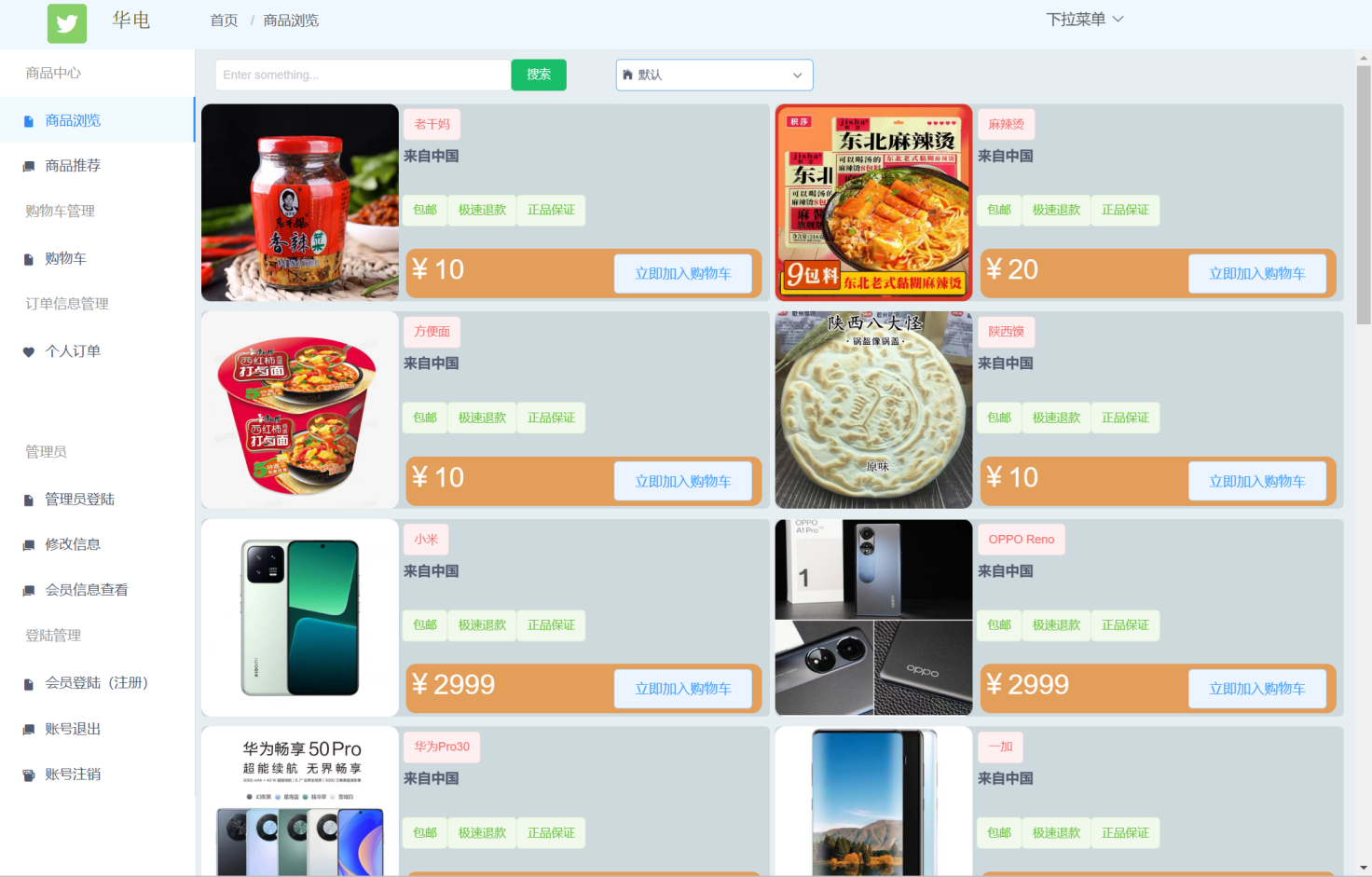
****

****

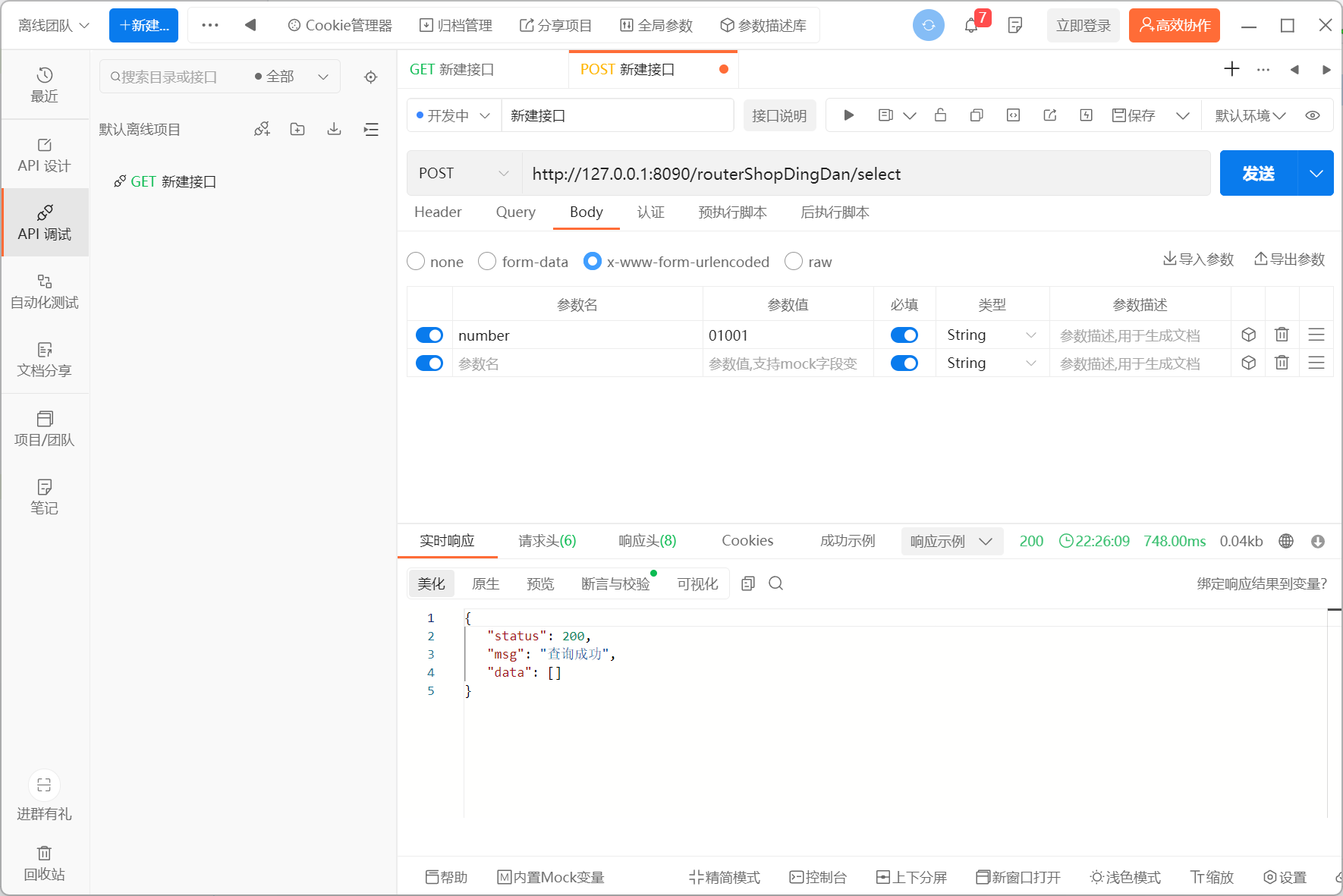
****

****

****

****

****

****

**1、使用技术**

**1.1 前端**

Vue+Vuex+router + axios + Element UI +Iview

Vue:前端框架

Vuex：Vue组件，全局数据存储与处理

router：Vue组件， Vue的路由技术

axios：基于ajax的JavaScript 异步请求库

Element UI：基于Vue的UI库

iview：基于Vue的UI库

**1.2 后端**

Node.js +MySQL + express

Node.js：JavaScript语言的后端运行库

MySQL ： nodejs的模块，操作mysql数据库

express： nodejs模块，服务器框架，用于接受请求数据，访问数据

1. **核心代码**

**2.1前端**

**axios请求**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | axios({  url: "http://127.0.0.1:8090/ShoppingCart/insert",  method: "POST",  data: {  Hno: this.$store.state.HeaderView.LoginStatus.user\_id, Sno: this.goods.id,  },  }).then(  (res) => {  console.log(res.data);  if(res.data.status === 200){  this.$message.success("添加成功");  }else {  this.$message.error("添加失败");  }  },  (err) => {  console.log(err);  this.$message.error("添加失败");  }  ); |

**Vue声明周期使用**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | if (  this.$store.state.HeaderView.LoginStatus.status &&  this.$store.state.HeaderView.LoginStatus.user === "manager" ) {  console.log()  } else {  this.$message.error("非管理员禁止操作!")  this.$router.push({  path: "/"  })  } |

**主要用途:**

**使用声明周期,在不同的生命周期判断当前用户权限,对于不同的操作实现不同的管理**

**Vue 入口函数**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | import Vue from 'vue'  import ElementUI from 'element-ui';  import 'element-ui/lib/theme-chalk/index.css';  import iview from 'iview'  import 'iview/dist/styles/iview.css'  import store from './store';  import VueRouter from 'vue-router'  import router from './router'; |
| import App from './App.vue' |
| // 关闭生产提示  Vue.config.productionTip = false  Vue.use(ElementUI);  Vue.use(iview)  Vue.use(VueRouter)  new Vue({  store,  router,  render: h => h(App),  }).$mount('#app') |

**Element UI的使用**

<div class="login\_view">

<el-form

:model="ruleForm"

:rules="rules"

ref="ruleForm"

label-width="100px"

class="demo-ruleForm"

>

<el-form-item label="用户名" prop="name">

<el-input v-model="ruleForm.name"></el-input>

</el-form-item>

<el-form-item label="密码" prop="password">

<el-input v-model="ruleForm.password" type="password"></el-input> </el-form-item>

</el-form>

<el-button

style="float: right; margin-left: 40px"

type="warning"

plain

@click="goToRegister"

>没有账号，注册</el-button

>

<el-button style="float: right" type="primary" @click="loginMember" >登陆</el-button

>

</div>

**iview的使用**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <Grid square  >  <GridItem v-for="(item,index) in pictures" :key="index">  <img :src="item" alt="" width="20%" height="20%"  style="border-radius: 15px;"  >  </GridItem>  </Grid> |

**2.2后端**

**nodejs的启动文件**

/\*\*

\* @name main

\* @author MateBook13

\* @description 2002

\* @date 2023/1/8

\*/

const express = require('express');

const routerOne = require('./Server/Router/router.js')

const routerMember = require('./Server/Router/routerMember.js') const routerManager = require('./Server/Router/routerManager.js') const routerCommodity = require('./Server/Router/routerCommodity.js')

const routerShoppingCart = require('./Server/Router/routerShoppingCart.js') const routerShopDingDan = require('./Server/Router/routerShopDingDan.js') const routerObject = require('./Server/Router/routerObject.js')

const cors = require('cors');

const bodyParser = require('body-parser');

const app = express();

app.use(bodyParser.urlencoded({

extended: false

}))

app.use(bodyParser.json())

app.use(cors())

app.use('/', routerOne)

app.use('/member', routerMember)

app.use('/manager', routerManager)

app.use('/routerCommodity', routerCommodity)

app.use('/ShoppingCart', routerShoppingCart)

app.use('/routerShopDingDan', routerShopDingDan)

app.use('/routerObject', routerObject)

app.use('/pictures',express.static('./Server/Temp'))//文件夹

app.listen(8090, function () {

console.log("RUNNING ! !");

})

**mysql数据库的操作**

const mysql = require('mysql')

// mysql 配置信息

const config = {

host: "rm-2ze81g5572t0v2231zo.mysql.rds.aliyuncs.com",

user: "messi",

password: "Xinan2002@",

port: "3306",

database: "mall\_system"//需要操作的数据库

};

const db = mysql.createPool(config)

let sql = `SELECT \*

FROM huiyuan;`;

db.query(sql, (err, results) => {

if (err) return console.log(err.message); res.send({

status: 200,

msg: "登陆成功",

data: {

status: "success", body: results

}

})

})

**express的使用**

**express通过路由实现对不同请求的具体划分**

app.use('/', routerOne)

app.use('/member', routerMember)

app.use('/manager', routerManager)

app.use('/routerCommodity', routerCommodity)

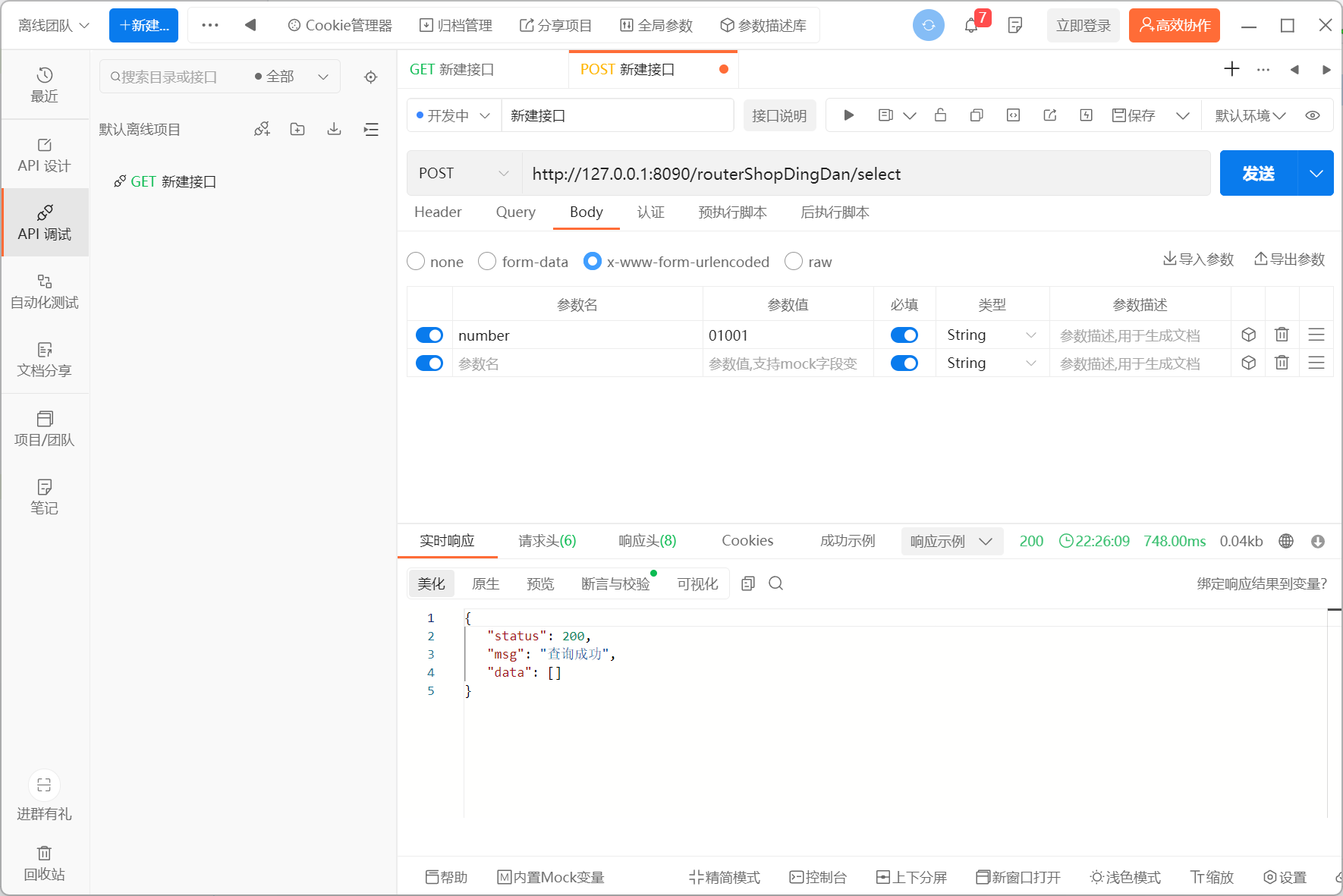
app.use('/ShoppingCart', routerShoppingCart)

app.use('/routerShopDingDan', routerShopDingDan)

app.use('/routerObject', routerObject)

**数据的模拟请求**

**使用 ApiPost 软件**



**数据库的操作**

**注册会员**

|  |  |
| --- | --- |
| 1  2  3 | -- 注册会员  INSERT INTO huiyuan  VALUES ('09010', '海贼', '21671', '12456464', '青海西宁'); |

**查询商品**

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6 | -- 查询商品  SELECT shangpin.Sno, SZno, Sname, Sprice, FROM shangpin | Sintro, ob.picture\_path  shangpin.Sno = ob.Sno; |
| INNER JOIN object\_pictures ob on |

**查询图片**

|  |  |
| --- | --- |
| 1  2  3  4 | -- 查询图片  SELECT picture\_path  FROM object\_pictures  WHERE Sno = '01001'; |

**模糊查询商品**

-- 模糊查询商品

SELECT \*

FROM shangpin

WHERE Sname like '%老%';

**查询购物车**

-- 查询购物车

SELECT \*

FROM shangpin

WHERE Sno IN (SELECT Sno

FROM gouwuche

WHERE Hno = '13206'

);

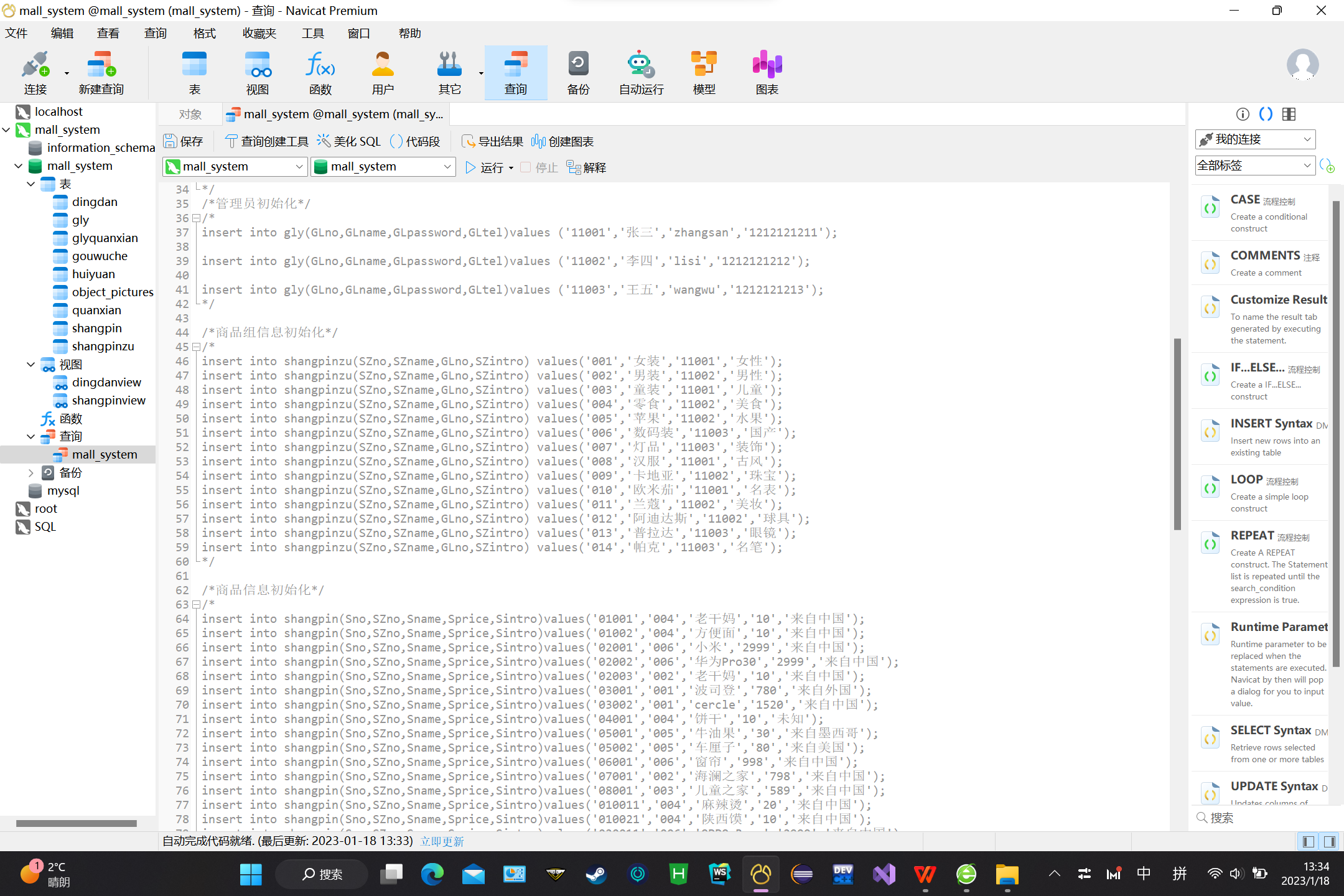
**添加订单**

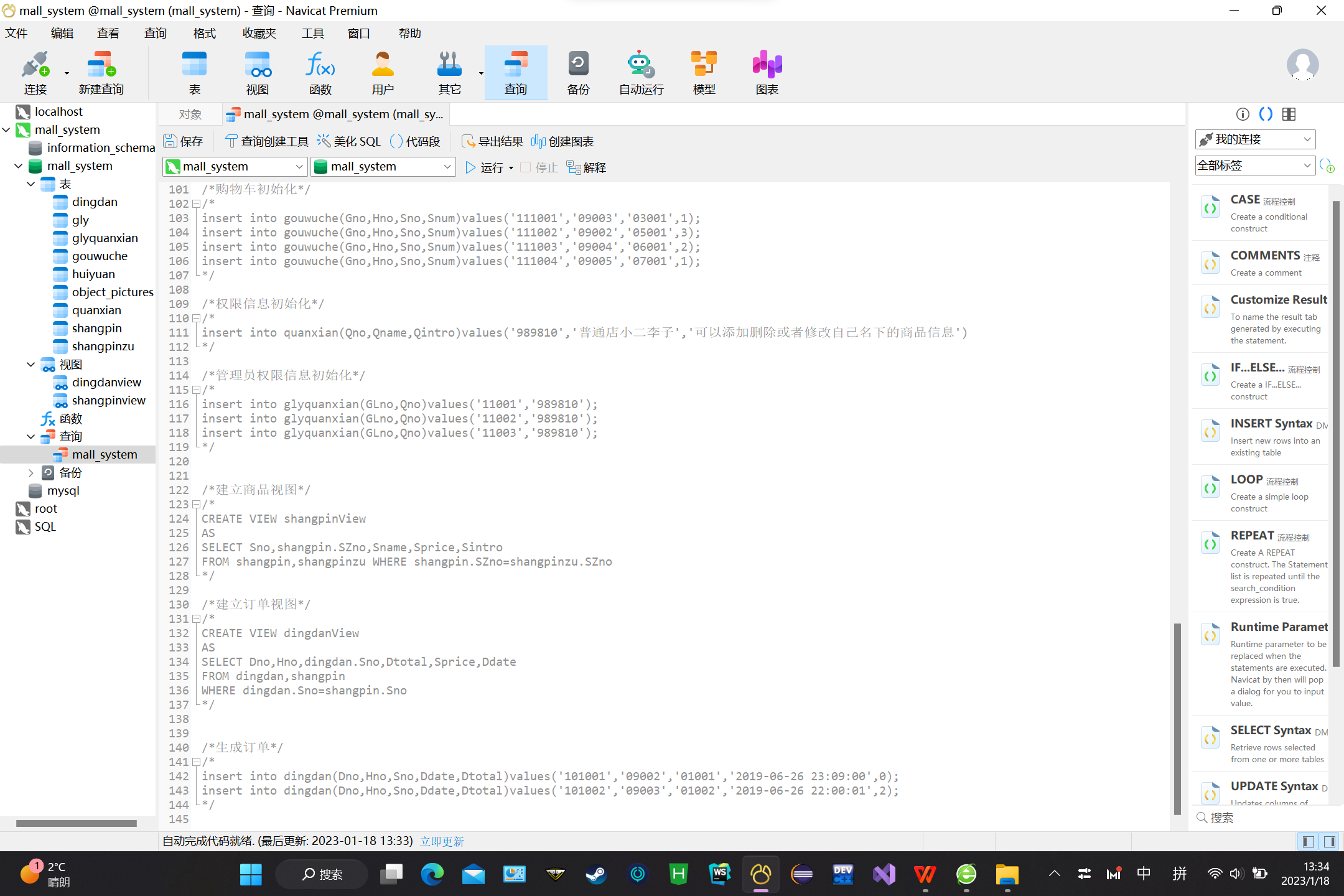
-- 添加订单

INSERT INTO dingdan (Hno, object\_id, Ddate, Dtotal)

VALUES ('13206','01001&01002',NOW(),345.78);

**5.2数据库设计及代码实现**

****

****

**/\*建立各表之间的联系\*/**

**/\***

**alter table 订单信息表**

**add constraint FK\_订单信息表\_REFRENCE\_会员注册信息表 foreign key (会员编号)**

**references 会员注册信息表(会员编号)**

**alter table 订单信息表**

**add constraint FK\_订单信息表\_REFRENCE\_商品信息表 foreign key (商品编号)**

**references 商品信息表(商品编号)**

**alter table 购物车信息表**

**add constraint FK\_购物车信息表\_REFRENCE\_会员注册信息表 foreign key (会员编号)**

**references 会员注册信息表 (会员编号)**

**alter table 购物车信息表**

**add constraint FK\_购物车信息表\_REFRENCE\_商品信息表 foreign key (商品编号)**

**references 商品信息表(商品编号)**

**alter table 商品组信息表**

**add constraint FK\_商品组信息表\_REFRENCE\_管理员信息表 foreign key (管理员编号)**

**references 管理员信息表(管理员编号)**

**alter table 商品信息表**

**add constraint FK\_商品信息表\_REFRENCE\_商品组信息表 foreign key (商品组编号)**

**references 商品组信息表(商品组编号)**

**alter table 管理员权限信息表**

**add constraint FK\_管理员权限信息\_REFRENCE\_权限信息表 foreign key (权限编号)**

**references 权限信息表(权限编号)**

**alter table 管理员权限信息表**

**add constraint FK\_管理员权限信息\_REFRENCE\_管理员信息表 foreign key (管理员编号)**

**references 管理员信息表 (管理员编号)**

**\*/**

**/\*管理员初始化\*/**

**/\***

**insert into gly(GLno,GLname,GLpassword,GLtel)values ('11001','张三','zhangsan','1212121211');**

**insert into gly(GLno,GLname,GLpassword,GLtel)values ('11002','李四','lisi','1212121212');**

**insert into gly(GLno,GLname,GLpassword,GLtel)values ('11003','王五','wangwu','1212121213');**

**\*/**

**/\*商品组信息初始化\*/**

**/\***

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('001','女装','11001','女性');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('002','男装','11002','男性');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('003','童装','11001','儿童');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('004','零食','11002','美食');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('005','苹果','11002','水果');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('006','数码装','11003','国产');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('007','灯品','11003','装饰');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('008','汉服','11001','古风');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('009','卡地亚','11002','珠宝');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('010','欧米茄','11001','名表');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('011','兰蔻','11002','美妆');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('012','阿迪达斯','11002','球具');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('013','普拉达','11003','眼镜');**

**insert into shangpinzu(SZno,SZname,GLno,SZintro) values('014','帕克','11003','名笔');**

**\*/**

**/\*商品信息初始化\*/**

**/\***

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('01001','004','老干妈','10','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('01002','004','方便面','10','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('02001','006','小米','2999','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('02002','006','华为Pro30','2999','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('02003','002','老干妈','10','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('03001','001','波司登','780','来自外国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('03002','001','cercle','1520','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('04001','004','饼干','10','未知');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('05001','005','牛油果','30','来自墨西哥');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('05002','005','车厘子','80','来自美国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('06001','006','窗帘','998','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('07001','002','海澜之家','798','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('08001','003','儿童之家','589','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('010011','004','麻辣烫','20','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('010021','004','陕西馍','10','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('020011','006','OPPO Reno','2999','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('020021','006','一加','2999','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('020031','002','橄榄菜','10','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('030011','010','欧家','780','来自外国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('030021','010','罗宾尼','1100','来自外国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('040011','004','比利时饼干','30','比利时');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('050011','005','青芒','30','来自越南');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('050021','005','靑椰','80','来自中国');**

**insert into shangpin(Sno,SZno,Sname,Sprice,Sintro)values('060011','006','床帘','998','来自中国');**

**\*/**

**/\*会员信息初始化\*/**

**/\***

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09002','李刚','223456','13029579891','南京');**

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09003','陈明','323456','13019579891','上海');**

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09004','陈志','423456','13719579891','天津');**

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09005','张力','523456','13729579891','深圳');**

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09006','王华','623456','13569579891','广州');**

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09007','洪溪','723456','13579579891','重庆');**

**insert into huiyuan(Hno,Hname,Hpassword,Htel,Hadd)values('09008','邓敏','823456','13589579891','成都');**

**\*/**

**/\*购物车初始化\*/**

**/\***

**insert into gouwuche(Gno,Hno,Sno,Snum)values('111001','09003','03001',1);**

**insert into gouwuche(Gno,Hno,Sno,Snum)values('111002','09002','05001',3);**

**insert into gouwuche(Gno,Hno,Sno,Snum)values('111003','09004','06001',2);**

**insert into gouwuche(Gno,Hno,Sno,Snum)values('111004','09005','07001',1);**

**\*/**

**/\*权限信息初始化\*/**

**/\***

**insert into quanxian(Qno,Qname,Qintro)values('989810','普通店小二李子','可以添加删除或者修改自己名下的商品信息')**

**\*/**

**/\*管理员权限信息初始化\*/**

**/\***

**insert into glyquanxian(GLno,Qno)values('11001','989810');**

**insert into glyquanxian(GLno,Qno)values('11002','989810');**

**insert into glyquanxian(GLno,Qno)values('11003','989810');**

**\*/**

**/\*建立商品视图\*/**

**/\***

**CREATE VIEW shangpinView**

**AS**

**SELECT Sno,shangpin.SZno,Sname,Sprice,Sintro**

**FROM shangpin,shangpinzu WHERE shangpin.SZno=shangpinzu.SZno**

**\*/**

**/\*建立订单视图\*/**

**/\***

**CREATE VIEW dingdanView**

**AS**

**SELECT Dno,Hno,dingdan.Sno,Dtotal,Sprice,Ddate**

**FROM dingdan,shangpin**

**WHERE dingdan.Sno=shangpin.Sno**

**\*/**

**三、课程设计总结或结论**

本次数据库课程设计我们做的是网上购物系统，随着互联网技术的高速发展，网上购物已是一种趋势，但是传统的销售模式已不再适应电商行业的经营模式，而这个网上购物系统可以给买家和卖家都提供着很大的便利。

我们从最开始的数据设计、需求分析开始，再到概念结构设计、E-R图描绘、模式的定义、物理结构设计等等。先前没有过数据库系统开发经验的我们开始一点点学习并将其付诸实践，从课本上模糊的概念到具体化的实现、从搭建数据库到前后端相连，我们体验了一个数据库系统的设计过程，亲自动手设计操作，对数据库的认识也有了进一步的提升。我们采用课堂之外的web前端进行开发，也是对自身的一次挑战，虽然开发过程中碰到了很多问题比如主键、触发器的设计、前端实现无法达到预期效果等等，我们通过查阅教科书、网上搜索解决方案、询问老师同学等方式解决了问题，收益颇丰。最终因为时间问题系统一些功能还未来得及完善，但是我们也新增了一些题目里面没有设计的新功能对系统进行了优化。

**四、参考文献**

[1] 萨师煊，王珊.数据库系统概论. 高等教育出版社, 第3版. 2000年2月