

《数据库系统》实验报告

实验名称：MIS 系统开发

成 绩：

专业班级：

姓 名：

学 号：

实 验 日 期：2018 年 05 月 18 日

实验报告日期：2018 年 05 月 24 日

一、实验目的

通过高级语言和嵌入式 sql 开发 MIS 系统，对 MIS 应用系统进行分析和数据库设计，绘制出 E-R/IDEFIX 图

二、实验内容

- 1) MIS 系统的题目和内容自选(如学生学籍管理系统，医疗档案管理系统，图书管理系统等等)。
- 2) 根据 MIS 系统的需求确定用到的实体，属性和联系。该系统至少包括 8 个实体和 7 个联系（包括一对一联系、一对多联系、多对多联系）。设计中体现关系完整性约束（包括主键约束、外键约束和空值约束）。
- 3) 有交互式界面，能通过界面插入、修改和删除数据，能够实现查询操作，包括连接查询、嵌套查询、分组查询。插入、删除操作能体现关系表的完整性约束，例如插入空值、删除有外键约束的数据时给出提示。

三、实验结果

电商平台 MIS 系统

卖家在电商平台管理自家商品信息，每件商品都有自己的历史购买信息与评价。顾客在平台浏览商品信息，并且管理与维护自己的购物车，购物车包含顾客中意的一些商品。顾客也可就某些商品下订单，订单可包含几种不同的订单项，由某种商品的属性如大小、颜色等，生成订单之后，顾客可支付此订单，完成一次购买行为。

列出所设计的数据库表结构

数据库表结构：

```
mysql> show tables;
+-----+
| Tables_in_shop |
+-----+
| hibernate_sequence |
| t_admin            |
| t_news            |
| t_order           |
| t_orderitem       |
| t_payment         |
| t_picture         |
| t_product         |
| t_product_t_picture |
| t_producttype     |
| t_user            |
| t_useraddress     |
+-----+
12 rows in set (0.00 sec)
```

建立 shop 数据库，共有 12 个 Table；

t_admin: 管理员信息，即商家信息；

t_news: 公告信息；

t_order: 订单信息；

t_orderitem: 订单项；

t_payment: 支付单；

t_picture: 网站图片信息记录；

t_product: 商品信息；

t_product_t_picture: 商品图片信息；

t_producttype: 商品类型；

t_user: 顾客信息；

t_useraddress: 顾客联系信息。

各 Table 具体包含属性见 ER 图；

简述程序实现的具体过程

1. 基于兴趣以及个人生活经验，确定开发 MIS 系统，电子商务平台。
2. 制作电子商务平台 ER 图，确定实体即相关必要属性；
3. 确定开发框架—Spring boot；

Spring boot 基于 Spring 框架开发，随着功能以及业务逻辑的日益复杂，应用伴随着大量的 XML 配置文件以及复杂的 Bean 依赖关系。随着 Spring 3.0 的发布，Spring IO 团队逐渐开始摆脱 XML 配置文件，并且在开发过程中大量使用“约定优先配置”（convention over configuration）的思想来摆脱 Spring 框架中各类繁复纷杂的配置（即时是 Java Config）。

Spring Boot 正是在这样的背景下被抽象出来的开发框架，它本身并不提供 Spring 框架的核心特性以及扩展功能，只是用于快速、敏捷地开发新一代基于 Spring 框架的应用程序。也就是说，它并不是用来替代 Spring 的解决方案，而是和 Spring 框架紧密结合用于提升 Spring 开发者体验的工具。同时它集成了大量常用的第三方库配置（例如 Jackson, JDBC, Mongo, Redis, Mail 等等），Spring Boot 应用中这些第三方库几乎可以零配置的开箱即用（out-of-the-box），大部分的 Spring Boot 应用都只需要非常少量的配置代码，开发者能够更加专注于业务逻辑。

4. 确定数据库层框架选择-JPA

JPA 的全称叫做 Java Persistence API，JPA 是一个基于 O/R 映射的标准规范，在这个规范中，JPA 只定义标准规则，不提供实现，使用者则需要按照规范中定义的方式来使用。目前 JPA 的主要实现有 Hibernate、EclipseLink、OpenJPA 等，事实上，由于 Hibernate 在数据访问解决技术领域的绝对霸主地位，JPA 的标准基本是由 Hibernate 来主导的。

5. 前端 UI 设计—Bootstrap
6. 依照 Web 开发的一般流程与规范，由于只有一名开发者，故先进行前端页面的开发，然后进行后端开发。

截屏给出程序界面，操作界面和操作结果

用户

1. 网站首页



2. 用户注册



3. 用户登录



4. 用户个人信息添加

小小商店 [首页](#) [商品](#) [公告](#) [购物车](#) Julian

[个人信息](#) [订单管理](#) [地址管理](#)

| 收件人 | 邮编 | 电话 | 地址 | 操作 |
|-------------|--------|-------------|-----------------|---------------------------------------|
| Jiliang Yan | 718600 | 18845788026 | Shanxi Province | 删除 修改 |

收货人

电话

邮编


地址

保存

© 小小商店 2018

5. 商品页面展示

小小商店 [首页](#) [商品](#) [公告](#) [购物车](#) Julian



美的不锈钢酸奶机

123

[查看](#) [购买](#)



高级餐具

3333

[查看](#) [购买](#)



红木茶具套装

1000

[查看](#) [购买](#)

共3条数据 << < 1 > >>

© 小小商店 2018

6. 商品详细信息

小小商店 [首页](#) [商品](#) [公告](#) [购物车](#) Julian



商品名称：美的不锈钢酸奶机

商品编码：2343434

商品型号：2a-dd-1s

商品价格：123

商品简介：超好用美的不锈钢酸奶机

加入购物车

© 小小商店 2018

7. 添加到购物车，查看购物车

小小商店

首页

商品

公告

购物车

Julian

| 名称 | 型号 | 编码 | 价格 | 数量 | 操作 |
|----------|----------|---------|-----|-------|----|
| 美的不锈钢酸奶机 | 2a-dd-1s | 2343434 | 123 | - 1 + | 删除 |

继续购买

清空购物车

结账

© 小小商店 2018

8. 下订单

小小商店

首页

商品

公告

购物车

Julian

Shanxi Provice
Jiliang Yan

地址

Shanxi Provice

邮编

718600

电话

18845788026

联系人

Jiliang Yan

| 名称 | 型号 | 编码 | 价格 | 数量 |
|----------|----------|---------|-----|----|
| 美的不锈钢酸奶机 | 2a-dd-1s | 2343434 | 123 | 1 |

回到购物车修改

确认

© 小小商店 2018

9. 订单管理

小小商店

首页

商品

公告

购物车

Julian

个人信息

订单管理

地址管理

| 订单日期 | 订单编号 | 订单金额 | 订单状态 | 收件人 | 操作 |
|-----------------------|----------------|--------|------|-------------|--|
| 2018-05-30 20:27:35.0 | 20180530202702 | 123.0 | 已取消 | Jiliang Yan | 查看 |
| 2018-05-30 20:37:14.0 | 20180530203719 | 2000.0 | 等待发货 | Jiliang Yan | 查看 取消订单 |
| 2018-05-30 20:48:19.0 | 20180530204881 | 4333.0 | 等待付款 | Jiliang Yan | 查看 付款 取消订单 |
| 2018-05-30 21:14:02.0 | 20180530211438 | 3456.0 | 已取消 | | 查看 |
| 2018-05-30 22:04:58.0 | 20180530220402 | 123.0 | 等待付款 | Jiliang Yan | 查看 付款 取消订单 |

共5条数据<<<1>>>

© 小小商店 2018

10. 支付

| | | | | | |
|-----------------------|----------------|-------|------|-------------|---|
| 2018-05-30 22:04:58.0 | 20180530220402 | 123.0 | 等待发货 | Jiliang Yan | 查看 取消订单 |
|-----------------------|----------------|-------|------|-------------|---|

共5条数据<<<1>>>

11. 支付单详情

小小商店

首页

商品

公告

购物车

Julian

地址

Shanxi Province

邮编

718600

电话

18845788026

联系人

Jiliang Yan

创建时间

2018-05-30 22:04:58.0

付款时间

发货时间

商品详情

| 名称 | 型号 | 编码 | 单价 | 数量 | 总价 |
|----------|----------|---------|-----|----|-----|
| 美的不锈钢酸奶机 | 2a-dd-1s | 2343434 | 123 | 1 | 123 |

© 小小商店 2018

卖家

1. 卖家登陆

管理员登陆

管理员

登陆

2. 商品管理

小小商店 首页 商品 公告

购物车 管理员

商品管理 订单管理 优惠管理 公告管理

| ID | 名称 | 型号 | 编码 | 价格 | 库存 | 操作 |
|----|----------|----------|---------|------|-----|----|
| 1 | 美的不锈钢酸奶机 | 2a-dd-1s | 2343434 | 123 | 3 | 修改 |
| 2 | 高级餐具 | df | sdfsdf | 3333 | 123 | 修改 |
| 3 | 红木茶具套装 | fddf | sdfdsf | 1000 | 222 | 修改 |

共3条数据 << < 1 > >>

添加

© 小小商店 2018

3. 订单管理

小小商店 首页 商品 公告

购物车 管理员

商品管理 订单管理 优惠管理 公告管理

| 订单日期 | 订单编号 | 订单金额 | 订单状态 | 收件人 | 操作 |
|-----------------------|----------------|--------|------|-------------|----------|
| 2018-05-30 20:27:35.0 | 20180530202702 | 123.0 | 已取消 | Jiliang Yan | 查看 |
| 2018-05-30 20:37:14.0 | 20180530203719 | 2000.0 | 等待发货 | Jiliang Yan | 查看 发货 删除 |
| 2018-05-30 20:48:19.0 | 20180530204881 | 4333.0 | 等待付款 | Jiliang Yan | 查看 删除 |
| 2018-05-30 21:14:02.0 | 20180530211438 | 3456.0 | 已取消 | | 查看 |
| 2018-05-30 22:04:58.0 | 20180530220402 | 123.0 | 等待发货 | Jiliang Yan | 查看 发货 删除 |

共5条数据 << < 1 > >>

© 小小商店 2018

4. 公告管理

小小商店

首页

商品

公告

购物车

管理员

商品管理

订单管理

优惠管理

公告管理

日期

公告标题

操作

2018-05-30 20:39:46.0

为庆祝数据库实验完成，优惠大促销

查看

修改

删除

共1条数据

<<

<

1

>

>>

添加

© 小小商店 2018

5. 商品信息修改

小小商店

首页

商品

公告

购物车

管理员

商品管理

订单管理

优惠管理

公告管理

商品名称

美的不锈钢酸奶机

商品价格

123

商品编码

2343434


商品型号

2a-dd-1s

商品库存

3

封面图片



选择文件

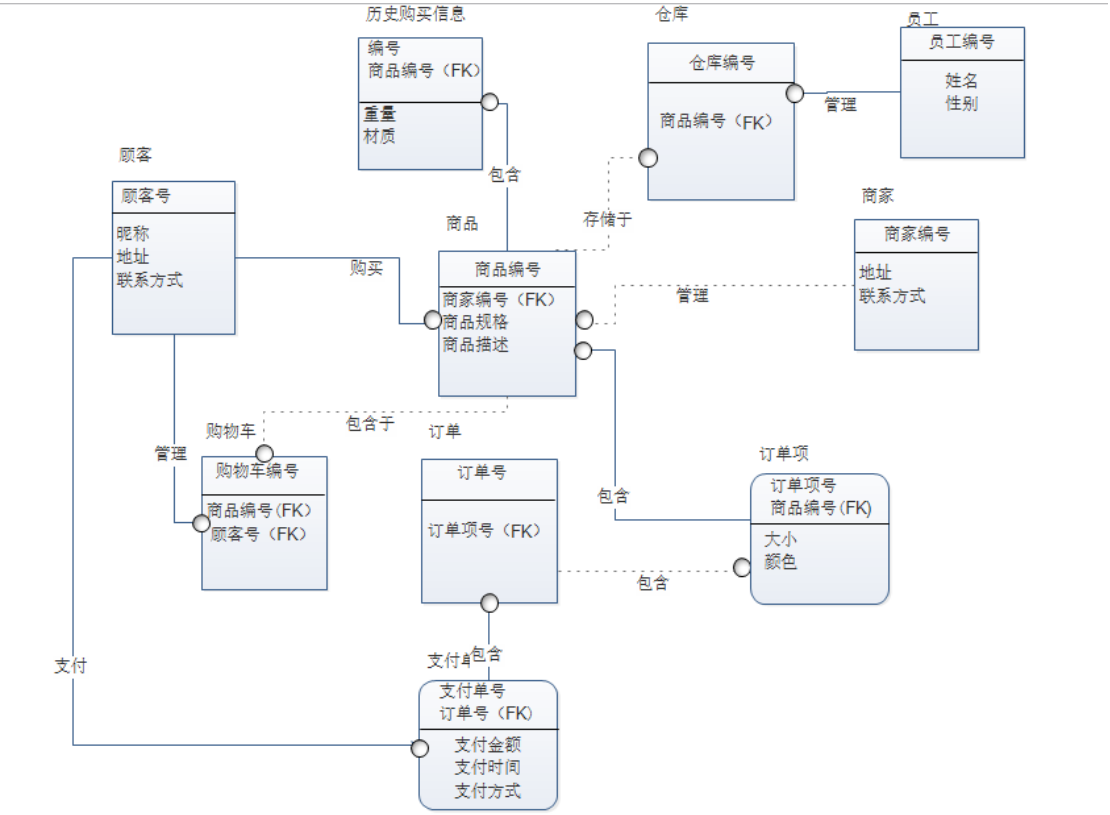
未选择任何文件

图片不能超过1MB

商品简介:

超好用美的不锈钢酸奶机

电商平台 ER 图



注：由于 ER 图是之前设计的，与系统实际实现在某些地方存在微小差别。

四、程序代码

主要代码：

```
package com.vitol6.shop.common;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

/**
```

```
* 全局配置类

* @author

* @version

*/

@Component

public class AppConfig {

    /**

    * 属性文件加载对象

    */

    private static PropertiesLoader propertiesLoader;

    @Value("${user.cookie.name}")

    public String USER_COOKIE_NAME;

    @Value("${user.cookie.age}")

    public int USER_COOKIE_AGE;

    /**

    * 获取配置

    */

    public static String getConfig(String key) {

        if (propertiesLoader == null){
```

```

        propertiesLoader = new
PropertiesLoader("application.properties");

    }

    return propertiesLoader.getProperty(key);

}

public static String getAdminPath() {

    return getConfig("adminPath");

}

public static String getFrontPath() {

    return getConfig("frontPath");

}

public static String getUrlSuffix() {

    return getConfig("urlSuffix");

}

}

/**

*

*/

package com.vitol6.shop.model;

import javax.persistence.*;

import java.io.Serializable;

/**

```

```

* 管理员

* @author * @date

*

*/

@Entity

@Table(name = "t_admin")

public class Admin implements Serializable {

    /**

    *

    */

    private static final long serialVersionUID = 1L;

    private Integer id;

    private String username;//账户名

    private String password;//密码

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    public Integer getId() {

        return id;

    }

    public void setId(Integer id) {

        this.id = id;

    }

```

```
    public String getUsername() {

        return username;

    }


    public void setUsername(String username) {

        this.username = username;

    }


    public String getPassword() {

        return password;

    }


    public void setPassword(String password) {

        this.password = password;

    }

}

/**
 *
 */

package com.vitol6.shop.service;


import com.vitol6.shop.model.Admin;

import com.vitol6.shop.repository.AdminRepository;
```

```
import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;


/**
 * @author Vito
 * @email zhouwentao16@gmail.com
 * @date 2013-7-9
 */
@Service
@Transactional
public class AdminService {

    @Autowired

    private AdminRepository adminDao;


    public boolean checkLogin(Admin admin) {

        return findByUsernameAndPassword(admin.getUsername(),
admin.getPassword()) == null ? false : true;

    }


    public Admin findByUsernameAndPassword(String username, String
password) {

        return adminDao.findByUsernameAndPassword(username, password);

    }
}
```



```
public void save(Admin admin) {  
    adminDao.save(admin);  
}  
  
public Admin findOne(Integer id) {  
    return adminDao.findOne(id);  
}  
  
}
```