

第一章 引言

1.1编写目的

数据库表的结构设计在整个项目开发当中是一个至关重要的环节，一个好的数据库设计是整个项目开发的良好开端。在后期开发中起到重要作用。本数据库设计说明书将从各方面来对我们的项目——**商易商品信息管理系统**的数据库模块进行整体的设计和规划，本说明书将从数据名称、数据域、数据数值等各方面进行严格规范，此数据库设计说明书将作为后续开发阶段的基础，是所有功能实现的基础，根据本数据库设计说明书可以使后续我们整个项目开发工作协调有序地进行。

1.2项目背景

软件名称：商易-商品信息管理系统

开发者：福州大学至诚学院2021级软件工程专升本1班“心满意组”[【心满意组博客园地址】](#)

本项目已经过前期的需求问卷调研得出结论：对于中小微量级商户而言，没有足够资金去定制装属软件使得他们不得不采用大量人力去管理商品信息。中小微量级商户迫切需要一款可以基本满足日程需求的商品信息管理系统。依靠计算机技术来管理商品仓库，可以对商品有一个直观的反馈，从而达到减少大量的人力、物力、减少管理成本、减轻员工劳动强度的目标，当管理人员面对市场变化可以快速做出调整和相应决策，使得整体的经营模式得到优化和改良。

本系统由本团队首次合作开发，在功能全面性上可能考虑的有所欠缺可能存在部分缺陷，但此系统开发及设计均为本团队原创，并可适用于大部分中小微量级的商户日程的商品信息的管理操作。

1.3项目范围

本系统主要建设范围包括：完成商户的登录登出功能，并记录登录次数，完成对商品入货/出货登记、商品信息、库存管理、供应商管理等模块的具体操作，管理员登录本系统可对商户的账户信息进行管理。

1.4预期读者

(1) **项目经理：**根据本《软件需求规格说明书》对本系统后期阶段做出整体设计以及管理。

(2) **开发人员：**根据本《软件需求规格说明书》对本系统进行整体的结构设计以及具体功能的实现。

(3) **测试人员：**根据本《软件需求规格说明书》对本系统进行软件测试流程，完成对系统整体实现功能的多方面测试，来测验是否达到预期要求。

(4) **最终用户**：该系统的使用者为中小微量级商户各自负责管理商品的操作人员，可根据本《软件需求规格说明书》了解和使用本系统的功能。

1.5使用工具

- EclipseIDE
- Navicat

1.6参考资料

- 数据库表结构设计的几条准则
<https://www.cnblogs.com/wyq178/p/8549715.html>
- 数据库表设计（一对多、多对多
<https://blog.csdn.net/fighteryang/article/details/82848505>
- 《构建之法》第三版 作者：邹欣

第二章 外部设计

2.1标识符和状态

数据库软件名称：Navicat

数据库使用语言：MySQL

序号	表名	描述信息	状态
1	g_stock	商品库存表，用来存储商品库存信息	暂时使用
2	p_details	进货明细表，用来记录进货单中的商品以及商品信息	暂时使用
3	p_registration	进货登记表，用来保存每次进货单的信息	暂时使用
4	refund	退货登记表，用来记录每次退货登记的信息	暂时使用
5	supplier	供应商信息登记表,用来记录每个供应商的信息	暂时使用
6	user	用户表，用来记录使用该系统的每个用户的信息	暂时使用

2.2 使用程序

访问此数据库的所有应用程序： 商易商品信息管理系统。

2.3标识符和约定

2.3.1标示符

字段时间格式：所有时间格式采用yyyy-mm-dd 的形式。

主外键约束：各表必须包含一个主键，外键引用正确。

命名格式：字段名和表名均以小写字母开头，遵循驼峰命名法。

命名约束：字段名和表名必须要见其名知其意。

2.3.2约定

本系统采用MVC模型，面向对象设计设计方法，使用Mybatis链接数据库将数据动态持久化存入数据库中。

2.4专门指导

无专门指导，在项目开始前部分成员有一些Web开发基础，各成员都有JAVA语言学习基础。

2.5支持软件

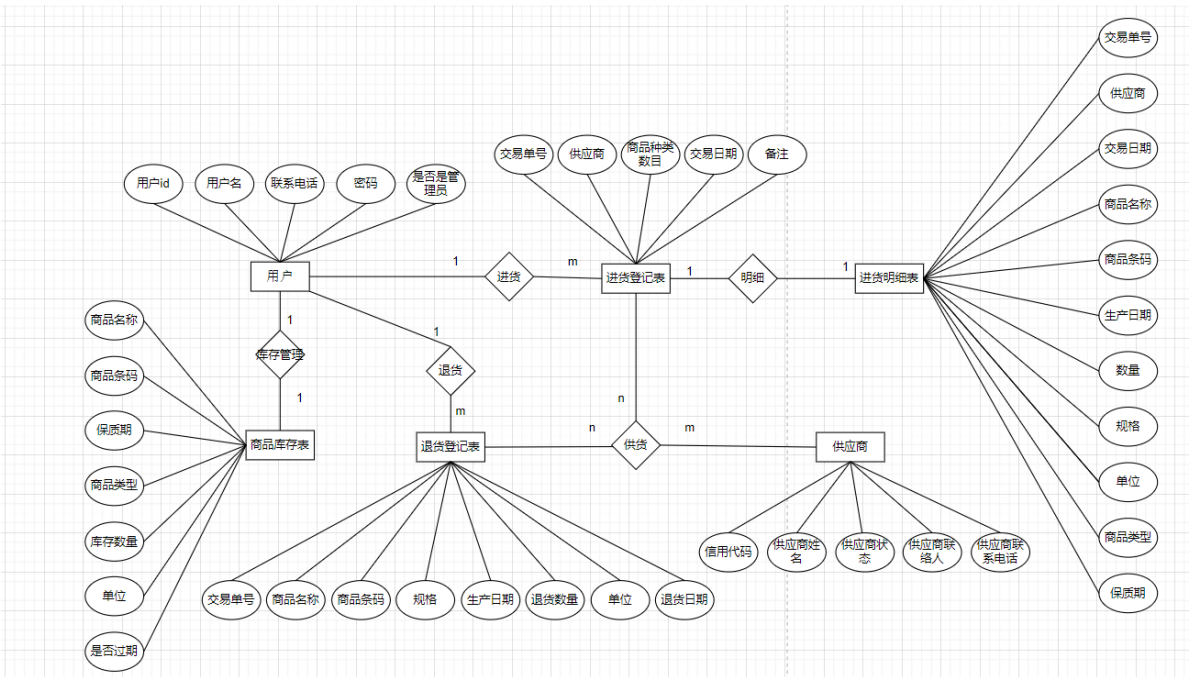
Web开发工具：EclipseIDE

数据库设计工具：Navicat

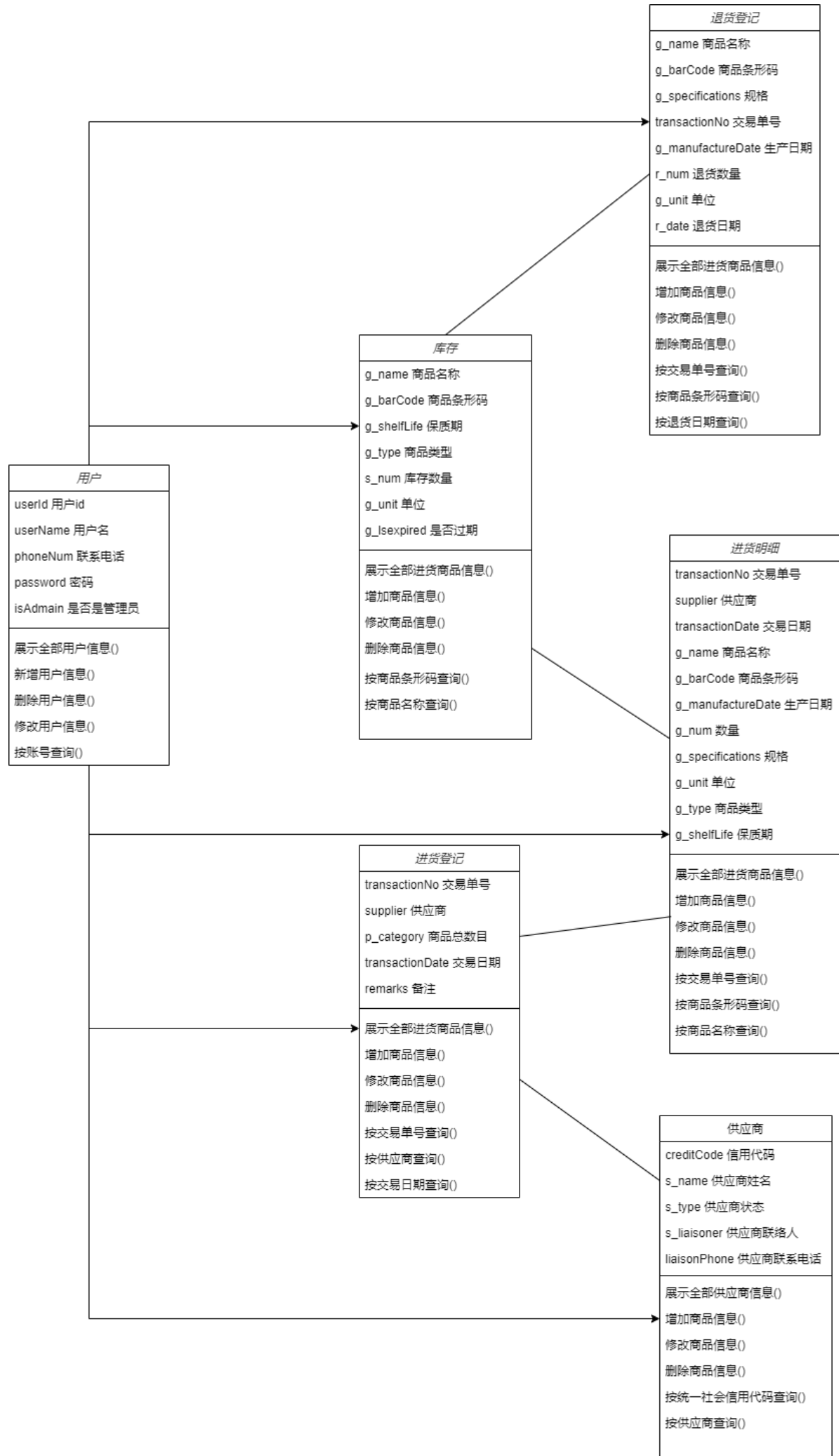
第三章 结构设计

3.1概念结构设计

3.1.1ER图



3.1.2类图



3.2逻辑结构设计

- g_stock

说明：商品库存表，用来存储商品库存信息。

对象 * g_stock @shangyi (mysql) - 表									
保存 添加字段 插入字段 删除字段 主键 上移 下移									
字段	索引	外键	触发器	选项	注释	SQL 预览			
名			类型	长度	小数点	不是 null	虚拟	键	注释
g_name			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		商品名称
g_barCode			varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	商品条形码
g_shelfLife			int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		保质期
g_type			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		商品类型
Is_num			int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		库存数量
g_nuit			varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		单位
g_lsexpired			int	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否过期

- p_details

说明：进货明细表，用来记录进货单中的商品以及商品信息。

对象 * p_details @shangyi (mysql) - 表									
保存 添加字段 插入字段 删除字段 主键 上移 下移									
字段	索引	外键	触发器	选项	注释	SQL 预览			
名			类型	长度	小数点	不是 null	虚拟	键	注释
transactionNo			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		交易单号
supplier			varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		供货商
transactionDate			date	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		交易日期
g_name			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		商品名称
g_barCode			varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	商品条形码
g_manufactureDate			date	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		生产日期
g_num			int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		数量
g_specifications			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		规格
g_unit			varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		单位
g_type			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		商品类型
g_shelfLife			int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		保质期

- p_registration

说明：进货登记表，用来保存每次进货单的信息。

对象 * p_registration @shangyi (mysql) - 表									
保存 添加字段 插入字段 删除字段 主键 上移 下移									
字段	索引	外键	触发器	选项	注释	SQL 预览			
名			类型	长度	小数点	不是 null	虚拟	键	注释
transactionNo			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	交易单号
supplier			varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		供应商名
p_category			int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		商品种类数目
transactionDate			date	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		交易日期
remarks			varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		备注

- refund

说明：退货登记表，用来记录每次退货登记的信息。

对象

* refund @shangyi (mysql) - 表

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
g_name	varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		商品名称
g_barCode	varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	商品条形码
g_specifications	varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		规格
transactionNo	varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		交易单号
g_manufactureDate	date	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		生产日期
r_num	int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		退货数量
g_unit	varchar	20	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		单位
r_date	date	6	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		退货日期

• supplier

说明：供应商信息登记表,用来记录每个供应商的信息。

对象

g_stock @shangyi (mysql) p_details @shangyi (mysql) p_registration @shangyi (mysql) refund @shangyi (mysql) supplier @shangyi (mysql) user @shangyi (mysql)

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
creditCode	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	统一社会信用代码
s_name	varchar	100	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		供应商姓名
s_type	varchar	50	0	<input type="checkbox"/>	<input type="checkbox"/>		供应商状态
s_liaisoner	varchar	50	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		联系人
liaisonPhone	varchar	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		联系电话

• user

说明：用户表，用来记录使用该系统的每个用户的信息。

对象

* user @shangyi (mysql) - 表

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
userId	int	5	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	用户id
userName	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		用户名
phoneNum	varchar	11	0	<input type="checkbox"/>	<input type="checkbox"/>		联系电话
password	varchar	25	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		密码
isAdmin	int	1	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		是否是管理员

3.3性能的规定

本系统在开发过程中将采用迭代式开发周期，开发过程中会尽量考虑系统整体的可维护性和可拓展性，在后续开发中若有需求会进行小范围内对个别数据的操作方式、数据精度等进行更改。

3.4故障处理要求

当系统出现网络崩溃、网页故障、系统崩溃等，可通过重新启动系统解决。系统重启后，将对数据库重新载入数据，进行回滚。若遇到不可回滚的系统操作，则需使用备用机上已存储的备份数据。

第四章 运用设计

4.1 系统安全和权限设计

4.1.1 系统安全

- 对使用本系统的用户进行和数据库信息进行比对完成身份验证。
- 定期对系统中的所有数据进行数据备份，防止由于意外操作、突发事件等的发生而导致用户数据丢失，定期备份可以保障系统的数据安全。
- 基于系统的安全规则和策略进行设计，同时需要降低用户操作错误导致的风险概率。

4.1.2 权限设计

- 数据库的权限和数据库的安全是息息相关的，不当的权限设置可能会导致各种各样的安全隐患，操作系统的某些设置也会对MySQL的安全造成影响。
- 需要界定权限的边界与关系，遵循一定的规则与用户进行关联。
- 系统对不同用户设置不同的权限，区分不同的用户：普通用户和管理员用户。

4.2 数据库实施

• g_stock

```
-- Table structure for g_stock
DROP TABLE IF EXISTS `g_stock`;
CREATE TABLE `g_stock` (
  `g_name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品名称',
  `g_barcode` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品条形码',
  `g_shelflife` int(0) NOT NULL COMMENT '保质期',
  `g_type` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品类型',
  `s_num` int(0) UNSIGNED NOT NULL COMMENT '库存数量',
  `g_nuit` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '单位',
  `g_isexpired` int(1) UNSIGNED ZEROFILL NOT NULL COMMENT '是否过期',
  PRIMARY KEY (`g_barcode`) USING BTREE,
  INDEX `s_g_nameKey` (`g_name`) USING BTREE,
  INDEX `s_g_typeKey` (`g_type`) USING BTREE,
  INDEX `s_g_unitKey` (`g_nuit`) USING BTREE,
  INDEX `s_g_shelflifeKey` (`g_shelflife`) USING BTREE,
  CONSTRAINT `s_g_barcodeKey` FOREIGN KEY (`g_barcode`) REFERENCES `p_details` (`g_barcode`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `s_g_nameKey` FOREIGN KEY (`g_name`) REFERENCES `p_details` (`g_name`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `s_g_shelflifeKey` FOREIGN KEY (`g_shelflife`) REFERENCES `p_details` (`g_shelflife`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `s_g_typeKey` FOREIGN KEY (`g_type`) REFERENCES `p_details` (`g_type`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `s_g_unitKey` FOREIGN KEY (`g_nuit`) REFERENCES `p_details` (`g_nuit`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;
```

• p_details

```
-- Table structure for p_details
DROP TABLE IF EXISTS `p_details`;
CREATE TABLE `p_details` (
  `transactionNo` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '交易单号',
  `supplier` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '供货商',
  `transactionDate` date NOT NULL COMMENT '交易日期',
  `g_name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品名称',
  `g_barcode` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品条形码',
  `g_manufactureDate` date NOT NULL COMMENT '生产日期',
  `g_num` int(0) NOT NULL COMMENT '数量',
  `g_specifications` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '规格',
  `g_unit` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '单位',
  `g_type` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品类型',
  `g_shelflife` int(0) NOT NULL COMMENT '保质期',
  PRIMARY KEY (`g_barcode`) USING BTREE,
  INDEX `supplierKey` (`supplier`) USING BTREE,
  INDEX `transactionDateKey` (`transactionDate`) USING BTREE,
  INDEX `transactionNoKey` (`transactionNo`) USING BTREE,
  INDEX `g_name` (`g_name`) USING BTREE,
  INDEX `g_unit` (`g_unit`) USING BTREE,
  INDEX `g_manufactureDate` (`g_manufactureDate`) USING BTREE,
  INDEX `g_specifications` (`g_specifications`) USING BTREE,
  INDEX `g_type` (`g_type`) USING BTREE,
  INDEX `g_shelflife` (`g_shelflife`) USING BTREE,
  CONSTRAINT `supplierKey` FOREIGN KEY (`supplier`) REFERENCES `supplier` (`s_name`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `transactionDateKey` FOREIGN KEY (`transactionDate`) REFERENCES `p_registration` (`transactionDate`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `transactionNoKey` FOREIGN KEY (`transactionNo`) REFERENCES `p_registration` (`transactionNo`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;
```

- p_registration

```
-- -----  
-- Table structure for p_registration  
-----  
DROP TABLE IF EXISTS `p_registration`;  
CREATE TABLE `p_registration` (  
  `transactionNo` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '交易单号',  
  `supplier` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '供应商名',  
  `p_category` int(0) NOT NULL COMMENT '商品种类数目',  
  `transactionDate` date NOT NULL COMMENT '交易日期',  
  `remarks` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL COMMENT '备注',  
  PRIMARY KEY (`transactionNo`) USING BTREE,  
  INDEX `transactionDate` (`transactionDate`) USING BTREE,  
  INDEX `r_supplierKey` (`supplier`) USING BTREE,  
  CONSTRAINT `r_supplierKey` FOREIGN KEY (`supplier`) REFERENCES `supplier` (`s_name`) ON DELETE RESTRICT ON UPDATE RESTRICT  
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;
```

- refund

```
-- -----  
-- Table structure for refund  
-----  
DROP TABLE IF EXISTS `refund`;  
CREATE TABLE `refund` (  
  `g_name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品名称',  
  `g_barCode` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '商品条形码',  
  `g_specifications` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '规格',  
  `transactionNo` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '交易单号',  
  `g_manufactureDate` date NOT NULL COMMENT '生产日期',  
  `r_num` int(0) NOT NULL COMMENT '退货数量',  
  `g_unit` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '单位',  
  `r_date` date NOT NULL COMMENT '退货日期',  
  PRIMARY KEY (`g_barCode`) USING BTREE,  
  INDEX `r_g_nameKey` (`g_name`) USING BTREE,  
  INDEX `r_g_specificationsKey` (`g_specifications`) USING BTREE,  
  INDEX `r_ntransactionNoKey` (`transactionNo`) USING BTREE,  
  INDEX `r_g_manufactureDateKey` (`g_manufactureDate`) USING BTREE,  
  INDEX `r_g_unitKey` (`g_unit`) USING BTREE,  
  CONSTRAINT `r_g_barCodeKey` FOREIGN KEY (`g_barCode`) REFERENCES `p_details` (`g_barCode`) ON DELETE RESTRICT ON UPDATE RESTRICT,  
  CONSTRAINT `r_g_manufactureDateKey` FOREIGN KEY (`g_manufactureDate`) REFERENCES `p_details` (`g_manufactureDate`) ON DELETE RESTRICT ON UPDATE RESTRICT,  
  CONSTRAINT `r_g_nameKey` FOREIGN KEY (`g_name`) REFERENCES `p_details` (`g_name`) ON DELETE RESTRICT ON UPDATE RESTRICT,  
  CONSTRAINT `r_g_specificationsKey` FOREIGN KEY (`g_specifications`) REFERENCES `p_details` (`g_specifications`) ON DELETE RESTRICT ON UPDATE RESTRICT,  
  CONSTRAINT `r_g_unitKey` FOREIGN KEY (`g_unit`) REFERENCES `p_details` (`g_unit`) ON DELETE RESTRICT ON UPDATE RESTRICT,  
  CONSTRAINT `r_ntransactionNoKey` FOREIGN KEY (`transactionNo`) REFERENCES `p_registration` (`transactionNo`) ON DELETE RESTRICT ON UPDATE RESTRICT  
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;
```

- supplier

```
-- -----  
-- Table structure for supplier  
-----  
DROP TABLE IF EXISTS `supplier`;  
CREATE TABLE `supplier` (  
  `creditCode` varchar(255) CHARACTER SET utf16 COLLATE utf16_general_ci NOT NULL COMMENT '统一社会信用代码',  
  `s_name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '供应商姓名',  
  `s_type` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NULL DEFAULT NULL COMMENT '供应商状态',  
  `s_liaisoner` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '联系人',  
  `liaisonPhone` varchar(11) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '联系电话',  
  PRIMARY KEY (`creditCode`) USING BTREE,  
  INDEX `s_name` (`s_name`) USING BTREE  
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;
```

- user

```
-- -----  
-- Table structure for user  
-----  
DROP TABLE IF EXISTS `user`;  
CREATE TABLE `user` (  
  `userId` int(0) NOT NULL AUTO_INCREMENT COMMENT '用户id',  
  `userName` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '用户名',  
  `phoneNum` varchar(11) CHARACTER SET utf8 COLLATE utf8_unicode_ci NULL DEFAULT NULL COMMENT '联系电话',  
  `password` varchar(25) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL COMMENT '密码',  
  `isAdmin` int(0) NOT NULL DEFAULT 0 COMMENT '是否是管理员',  
  PRIMARY KEY (`userId`) USING BTREE  
) ENGINE = InnoDB AUTO_INCREMENT = 2 CHARACTER SET = utf8 COLLATE = utf8_unicode_ci ROW_FORMAT = Dynamic;  
  
SET FOREIGN_KEY_CHECKS = 1;
```

第五章 数据库验证验收标准

序号	基本功能	功能实现
1	数据的查询与提取	根据MySQL语言需求查询属性、元组等，查询结果提取数据
2	数据的实时更新	可以动态更新数据库中的数据
3	数据的增删改	根据MySQL语言能对数据库中的数据进行相应操作

