**项目设计报告**

**19智能班 赵炳臣、韩旭、朱施宇**

目录

[一、数据库逻辑设计 2](#_Toc55668486)

[1.E-R图 2](#_Toc55668487)

[2.数据字典 2](#_Toc55668488)

[3.触发器 4](#_Toc55668489)

[二、数据库物理设计 5](#_Toc55668490)

[1.外键索引 5](#_Toc55668491)

[2.异常处理机制 5](#_Toc55668492)

[3.视图安全 5](#_Toc55668493)

[4.数据隐私安全 5](#_Toc55668494)

[5.数据保存安全 6](#_Toc55668495)

[6.数据修改安全 6](#_Toc55668496)

[7.其他 6](#_Toc55668497)

[三、应用程序设计 7](#_Toc55668498)

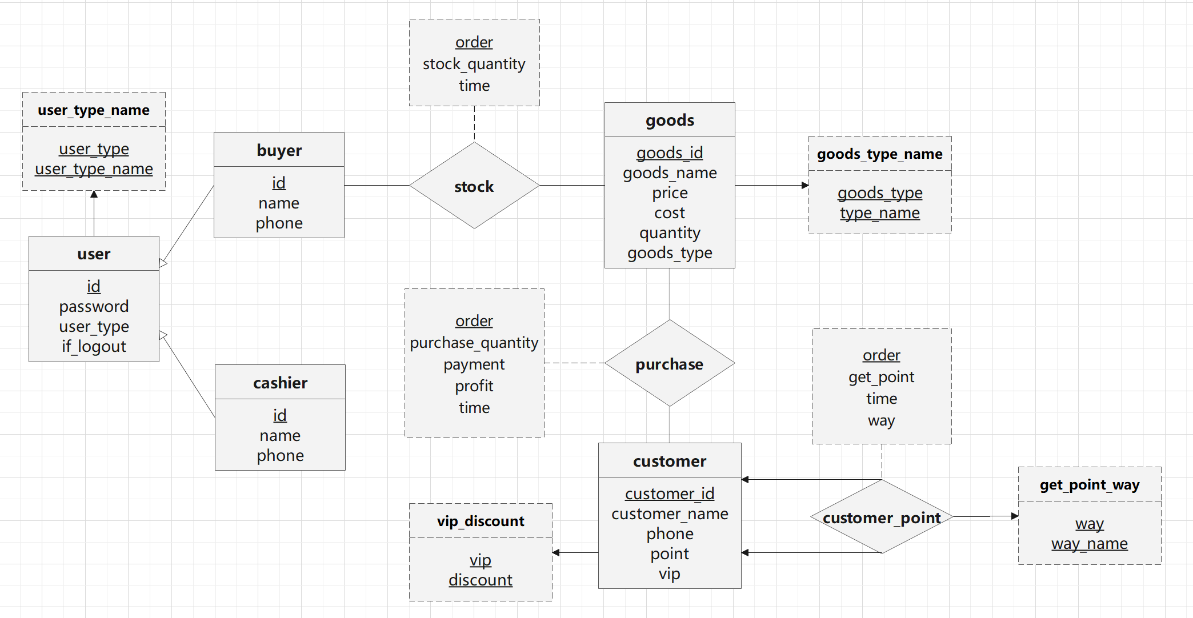
[1.功能设计 7](#_Toc55668499)

[2.界面设计 8](#_Toc55668500)

[3.事务设计 9](#_Toc55668501)

# 一、数据库逻辑设计

## 1.E-R图

****

## 2.数据字典

**user**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| id | int | √ | √ | √ | √ |  |
| password | varchar(10) |  | √ |  |  |  |
| user\_type\_name | int |  | √ |  |  |  |
| if\_logout | int |  | √ |  |  | 0 |

**cashier**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| id | int | √ | √ | √ |  |  |
| name | varchar(20) |  |  |  |  |  |
| phone | varchar(20) |  |  |  |  |  |

**buyer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| id | int | √ | √ | √ |  |  |
| name | varchar(20) |  |  |  |  |  |
| phone | varchar(20) |  |  |  |  |  |

**customer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| customer\_id | int | √ | √ | √ | √ |  |
| customer\_name | varchar(20) |  |  |  |  |  |
| phone | varchar(20) |  |  |  |  |  |
| point | decimal(10,2) |  | √ |  |  | 0 |
| vip | int |  | √ |  |  | 0 |

**goods**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| goods\_id | int | √ | √ | √ | √ |  |
| goods\_name | varchar(20) |  | √ |  |  |  |
| price | decimal(10,2) |  | √ |  |  |  |
| cost | decimal(10,2) |  | √ |  |  |  |
| quantity | int |  | √ |  |  | 0 |
| goods\_type | int |  | √ |  |  | 0 |

**parchase**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| customer\_id | int |  | √ |  |  |  |
| goods\_id | int |  | √ |  |  |  |
| quantity | int |  | √ |  |  |  |
| time | timestamp |  | √ |  |  | current |
| payment | decimal(10,2) |  | √ |  |  | 0.00 |
| profit | decimal(10,2) |  | √ |  |  | 0.00 |
| order | int | √ | √ | √ | √ |  |

**stock**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| order | int | √ | √ | √ | √ |  |
| goods\_id | int |  | √ |  |  |  |
| id | int |  | √ |  |  |  |
| quantity | int |  | √ |  |  |  |
| time | timestamp |  | √ |  |  | current |

**customer\_point**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| order | int | √ | √ | √ | √ |  |
| customer\_id | int |  | √ |  |  |  |
| get\_point | decimal(10,2) |  | √ |  |  |  |
| time | timestamp |  | √ |  |  | current |
| way | int |  | √ |  |  | 1 |

**vip\_discount**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| vip | int | √ | √ | √ |  |  |
| discount | decimal(10,2) | √ | √ | √ |  |  |

**user\_type\_name**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| user\_type | Int | √ | √ | √ |  |  |
| user\_type\_name | varchar(20) | √ | √ | √ |  |  |

**goods\_type\_name**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| goods\_type | int | √ | √ | √ |  |  |
| type\_name | varchar(20) | √ | √ | √ |  |  |

**get\_point\_way**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Datatype | PK | NN | UQ | AI | Default |
| way | int | √ | √ | √ |  |  |
| way\_name | varchar(20) | √ | √ | √ |  |  |

## 3.触发器

**1)**user表insert after触发器：

当有新数据插入本表后，会自动根据新元组的user\_type数值，向casier表或buyer表插入新数据{id=new.id,name=null,phone=null}。

**2)**purchase表insert after触发器：

当有新数据插入本表后，会使goods表中的对应商品扣除新数据的数量；会在customer\_point表中插入新记录，记录由本次交易顾客所获得的point值。

**3)** customer\_goods表insert after触发器：

当有新数据插入本表后，会自动增加customer表中对应顾客的point值，并根据更新之后的point值，更新用户的vip等级。

**4)** stock表insert after触发器：

当有新数据插入本表后，会使goods表中的对应商品增加新数据quantity的数量。

# 二、数据库物理设计

## 1.外键索引

|  |  |  |  |
| --- | --- | --- | --- |
| **表名** | **属性** | **参考表名** | **参考属性** |
| **user** | **user\_type** | **user\_type\_name** | **user\_type** |
| **cashier** | **id** | **user** | **id** |
| **buyer** | **id** | **user** | **id** |
| **customer** | **vip** | **vip\_discount** | **vip** |
| **goods** | **goods\_type** | **goods\_type\_name** | **goods\_type** |
| **purchase** | **customer\_id** | **customer** | **customer\_id** |
| **purchase** | **goods\_id** | **goods** | **goods\_id** |
| **stock** | **id** | **buyer** | **id** |
| **stock** | **goods\_id** | **goods** | **goods\_id** |
| **customer\_point** | **customer\_id** | **customer** | **customer\_id** |
| **customer\_point** | **way** | **get\_point\_way** | **way** |

## 2.异常处理机制

本系统提供若干异常处理机制，能够捕获由各种错误引发的异常，例如输入数据类型不符合对应数据库要求、查询错误以及数据不合法等问题。

在用户注册、登陆、修改密码活动等提交操作时，系统会对对数据合法性进行判断，若数据不合法则会返回错误信息并提醒用户重新输入。

## 3.视图安全

本系统共存在三种用户角色：管理员、收银员、进货员。不同的账号拥有不同的角色，不同的角色在使用本系统时会进入不同的操作界面，而账号密码不匹配或账号已被注销者则无法进入任何操作界面。

## 4.数据隐私安全

管理员拥有数据库的最高管理权力，可以查看和统计任何记录和信息，并在必要时可以修改用户信息、商品信息和顾客信息，但不可以修改任何交易记录、进货记录。而且管理员无法查看其它用户的密码等隐私数据。

收银员只能在其界面进行顾客的登记与交易，无法在其界面查看与修改其他信息。

进货员只能在其界面进行商品的登记与购进，无法在其界面查看与修改其他信息。

系统通过限制不同操作界面的功能和权限，保证了数据的隐私安全。

## 5.数据保存安全

本系统在保存数据之前会先对数据的合法性进行判断，对于不合法数据，对应的数据库将不会进行任何改动，并将不合法信息返回给操作界面，要求用户重新进行输入数据。

系统所有操作界面中均不存在对数据库的删除操作，仅管理员拥有对实体信息的修改权限，从而确保所有实体数据和记录数据均能够稳定的保存在数据库中。

本系统数据库会定期进行备份，以确保数据不会因突发情况而丢失。

## 6.数据修改安全

本系统数据库设置多个触发器，以确保当一张数据表中的某些数据发生改动时，与之关联的数据表能够及时准确地随之修改。

## 7.其他

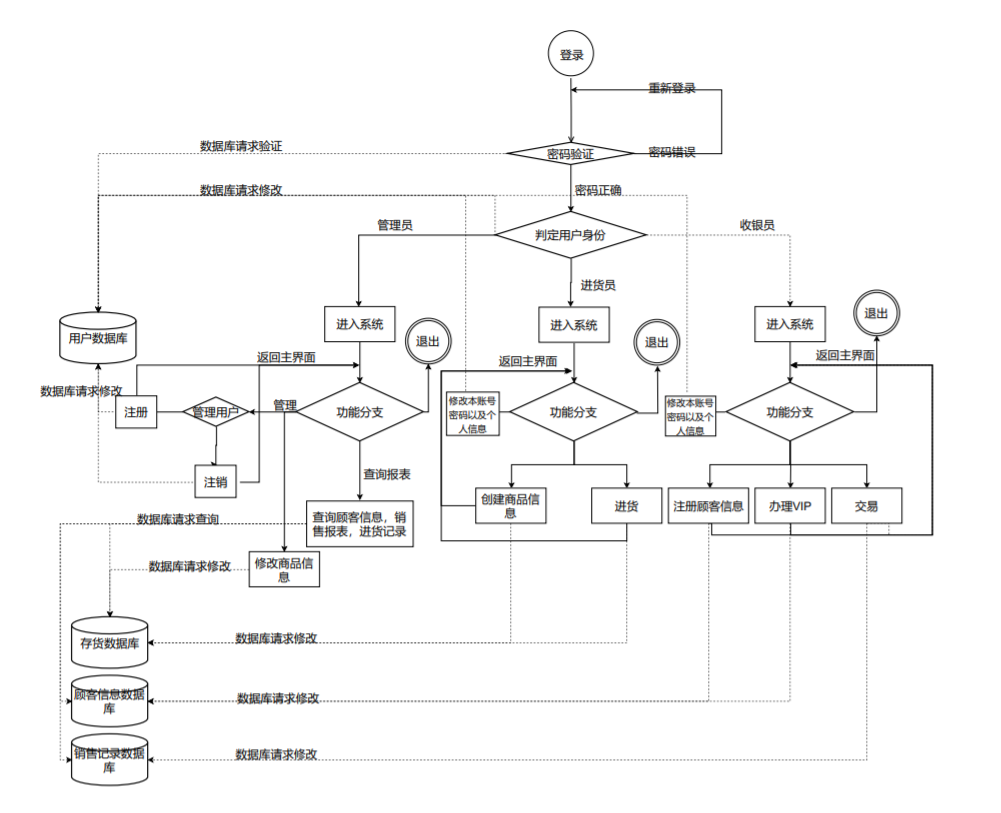
为了提升数据库的稳定性且提升效率，我们将表进行了拆解，并针对表中的主键及外键建立了索引，从而提升查询的效率。

在规范化的过程中，我们更多的采用了数据库中的外键约束，对于允许为空的属性在关联删除时选择置空，不允许为空的则选择级联删除，从而维护了数据库的一致性，避免不合法的数据存在数据库当中。

# 三、应用程序设计

## 1.功能设计

功能流程图：

****

功能描述：

**登录：**

**1）**用户的账号：向服务器发送请求，完成登录，并且确定登录用户的身份。

**管理员：**

**1）**管理系统用户的账号：

注册：管理员可以注册不同的新用户类型，这些新的用户根据类型不同拥有不同的数据访问权限。

（1）注册管理员账号，新的管理员账号将会具有管理员的权限，可对其他用户进行注册与注销，同时对全体数据具有查询权限。

（2）注册收银员账号，新的收银员账号将会具有修改自身密码，管理交易，管理自身信息的权限，并且可以管理顾客信息与VIP信息。

（3）注册进货员信息，新的进货员账号将会有创建新商品类别以及在已有商品的基础上继续进货的权限并且可以修改自身的信息。

注销：管理员可以注销任意用户类型。注销后，用户失去所有权限，并且不再能登录系统。但是用户的信息会作为历史记录被保留。

**2）**查看顾客的信息：

可以查询历史上所有用户的联系方式、购买信息。

**3）**查看进货情况：

管理员可以查询历史上所有的进货记录。

**4）**查看销售情况：

管理员可以查询自定义起始时间与终止时间之间，所有商品的销售情况，并且针对每种商品，显示商品的总销售额与总利润。也可以查询这段时间内所有商品的总利润。管理员可以选择按照总销售额或者总利润为标准对各种商品的销售情况进行排序查看。

**5）**查询VIP用户的消费情况：

管理员可以以用户id为分组，按照VIP等级降序的方式查看VIP最高等级的顾客的购买记录。

**收银员：**

**1）**修改自身账号密码：

收银员可以在系统内修改自身的账号密码。

**2）**修改自身的联系方式与个人信息：

收银员可以在系统内修改自身的联系方式与个人信息。

**3）**注册客户信息：

收银员端的系统如果发现了新的顾客，可以在系统内注册这个新用户的信息，形成新的顾客信息。

**4）**交易：

当顾客完成了一次消费，收银员具有权限，将交易记录提交给系统，并且由系统根据VIP等级自动生成折扣系数。

**5）**办理VIP：

收银员具有权限，可以为用户单独购买VIP权限。完成后可以向系统提交。

**进货员：**

**1）**修改自身账号密码：

进货员可以在系统内修改自身的账号密码。

**2）**修改自身的联系方式与个人信息：

进货员可以在系统内修改自身的联系方式与个人信息。

**3）**创建商品信息：

进货员可以创建新商品类别。

**4）**进货：

进货员可以在已有商品的基础上继续进货。

## 2.界面设计

程序包括一个入口登陆界面，三个主界面以及多个嵌套界面。

**1）入口登陆界面**：可以实现登录功能，确定用户类型。

**2）管理员界面**：可实现功能——管理系统用户的账号、查看顾客的信息、查看进货情况、查看销售情况、查询VIP用户的消费情况。

**3） 收银员界面**：可实现功能——修改自身账号密码、修改自身的联系方式与个人信息、注册客户信息、交易、办理VIP。

## 3.事务设计

**收银员事务**

**1）**修改密码：输入旧密码，在user表中查询验证，输入新密码，验证成功后在user表中对应id更新password。

**2）**修改个人身份信息：输入新的姓名、联系方式，在cashier表中对应id更新name、phone。

3）注册顾客信息：输入顾客姓名、联系方式，向customer表中的customer\_name、phone属性插入数据（其余属性会默认生成），显示新顾客的customer\_id。

**4）**交易：输入顾客id、商品id、商品数量进行交易，显示顾客支付金额，更新商品数量，留下交易记录。

(1)本操作应当在cashier（收银员）的界面完成。

(2)收银员在进行一次交易时，只需要将customer\_id、goods\_id、quantity三个数据填入系统，即可完成此次交易操作。内部程序的实现过程为：

利用customer\_id查出discount；

利用goods\_id查出price和cost；

利用quantity、discount、price计算出payment；

利用quantity、cost、payment计算出profit；

将{customer\_id, goods\_id, quantity, payment, profit}插入到purchase表中；

purchase表触发器自动实现商品数量的扣除以及顾客point的获得记录。

**5）**顾客直接购买vip点：输入顾客id、需购买vip点数，更新用户vip点数，留下购买记录。

(1) 本操作应当在cashier（收银员）的界面完成。

(2)在顾客直接购买vip点时，收银员需要将customer\_id、get\_point填入系统，即可完成该操作。内部程序的实现过程为：

向customer\_point表中插入新数据{customer\_id, get\_point, way=2}；

在customer\_point表的触发器自动实现customer表中顾客point和vip值的更新。

**进货员事务**

**1）**修改密码：输入旧密码，在user表中查询验证，输入新密码，验证成功后在user表中对应id更新password。

**2）**修改个人身份信息：输入新的姓名、联系方式，在cashier表中对应id更新name、phone。

**3）**创建商品信息：输入商品名称、售价、成本、类型，向goods表中的goods\_name、price、cost、goods\_type属性插入数据（其余属性会默认生成），显示新商品的goods\_id。

**4）**进货：输入商品id、进货数量，更新商品数量，留下进货记录。

(1)本操作应当在buyer（进货员）的界面完成。

(2)在进货时，进货员需要将goods\_id、quantity填入系统中，即可完成该操作。内部程序的实现过程为：

获取进货员自己的id；

向stock表中插入新数据{goods\_id, id, quantity}；

在stock表的触发器自动实现goods表中商品数量的增加。

**管理员事务**

**1）**注册：输入初始密码、用户类型，向user表中的password、user\_type属性插入数据（其余属性会默认生成），显示新用户id。

**2）**注销/解除注销：输入用户id，在user表中对应id更新if\_logout的值。

**3）**查看顾客信息：直接查询cutomer表，输出顾客的所有信息。

**4）**查询（一定时间限度内）所有用户的消费额：给出start\_time和end\_time，从customer表中查询出所有满足。end\_time > time > start\_time的条目，然后用order by vip,customer\_id对用户数据进行分组与排序。

**5）**查询（一定时间限度内）所有商品的销量、利润，查询各类商品的销售情况：给出start\_time和end\_time，从purchase表中查询出所有满足end\_time > time > start\_time的条目，然后用goods\_id对各个商品的销售记录进行sum(payment),sum(profit)聚类,最后通过不同的参数确定排序策略——选择order by sum(payment) desc 或者是 order by sum(profit) desc。

**6）**查询所有商品的进货记录： 直接查询stock表，获得所有信息。

**7）**修改商品数据：通过goods\_id对goods表中的信息进行定向修改。