

**Coupay** 系统设计文档系列

# 数据库设计说明书

<b>1 引言 .....</b>	<b>3</b>
1.1 编写目的.....	3
1.2 背景.....	3
1.3 定义.....	3
1.4 参考资料.....	4
<b>2 外部设计 .....</b>	<b>4</b>
2.1 标识符和状态.....	4
2.2 使用它的程序.....	4
2.3 约定.....	4
2.4 支持软件.....	4
<b>3 结构设计 .....</b>	<b>4</b>
3.1 概念结构设计.....	4
3.1.1 局部视图.....	4
3.1.2 综合视图.....	9
3.2 逻辑结构设计.....	9
3.3 物理结构设计.....	13
<b>4 运用设计 .....</b>	<b>14</b>
4.1 数据字典设计.....	14
4.2 安全保密设计.....	17

# 1 引言

## 1.1 编写目的

本说明书说明系统的数据库设计、结构情况，用于开发人员进行项目设计，同时也为后续的数据库维护工作提供了良好的使用说明，也可以作为未来版本升级时的重要参考资料。

## 1.2 背景

数据库名称: CoupayDatabase

使用此数据库的软件系统: 移动支付系统 Coupay.

系统开发项目任务提出者: 花旗银行。

用户: 主要用户为系统开发人员和系统管理员。

所使用 DBMS: MySQL 社区版 5.5.29

## 1.3 定义

注册者: 注册 Coupay 系统的用户，包括商家和客户。

账户: 用户注册后系统自动创建的一个账户，用户可用这个用户进行存款，转账。

绑定银行账户: 用户注册后可以把在 Coupay 系统合作银行的银行账户进行绑定，以方便存款，转账。

转账: 注册用户可以从自己的账户给其他账户转账，也可以从绑定银行账户给其他账户转账。

消费商品信息: 若客户的转账是用于购买商品，则用户可以查看这次转账包含的商品信息。另外，商品信息还用于 Coupay 系统的数据挖掘。

优惠券: 合作商家可以发布优惠券，客户可收集优惠券，优惠券有折扣，抵价等类型，客户可用优惠券到发布商家消费，也可以把优惠券送给好友。

VIP 卡: 合作商家可以发布 VIP 卡，客户可以申请 VIP 卡，获得消费上的优惠。

消费分享: 客户购买商品完成转账后可以把这次消费经历分享给圈里的好友，评论、推荐购买的商品。

好友: 客户可以与其他使用 Coupay 系统的客户成为好友，成为好友后可以看到彼此的消费分享, 可以互相赠送优惠券。

关注商家: 注册客户可以通过 Coupay 系统关注商家，关注商家后，可以查看商家的动态，有机会获得商家的优惠券，有机会成为商家的会员。

通知消息: 在某些情况 Coupay 系统需要向注册用户发送的通知消息，例如收到转账的通知。

## 1.4 参考资料

软件需求说明书.doc

# 2 外部设计

## 2.1 标识符和状态

数据库代码: CoupayDatabase

用户: admin

密码: admin

状态: 开发模式。

## 2.2 使用它的程序

本数据库专门为移动支付系统 Coupay 设计。

## 2.3 约定

数据项名称采用驼峰式表示，字符集采用 UTF-8。

## 2.4 支持软件

MySQL community Server 5.5.29.

MySQL Workbench.

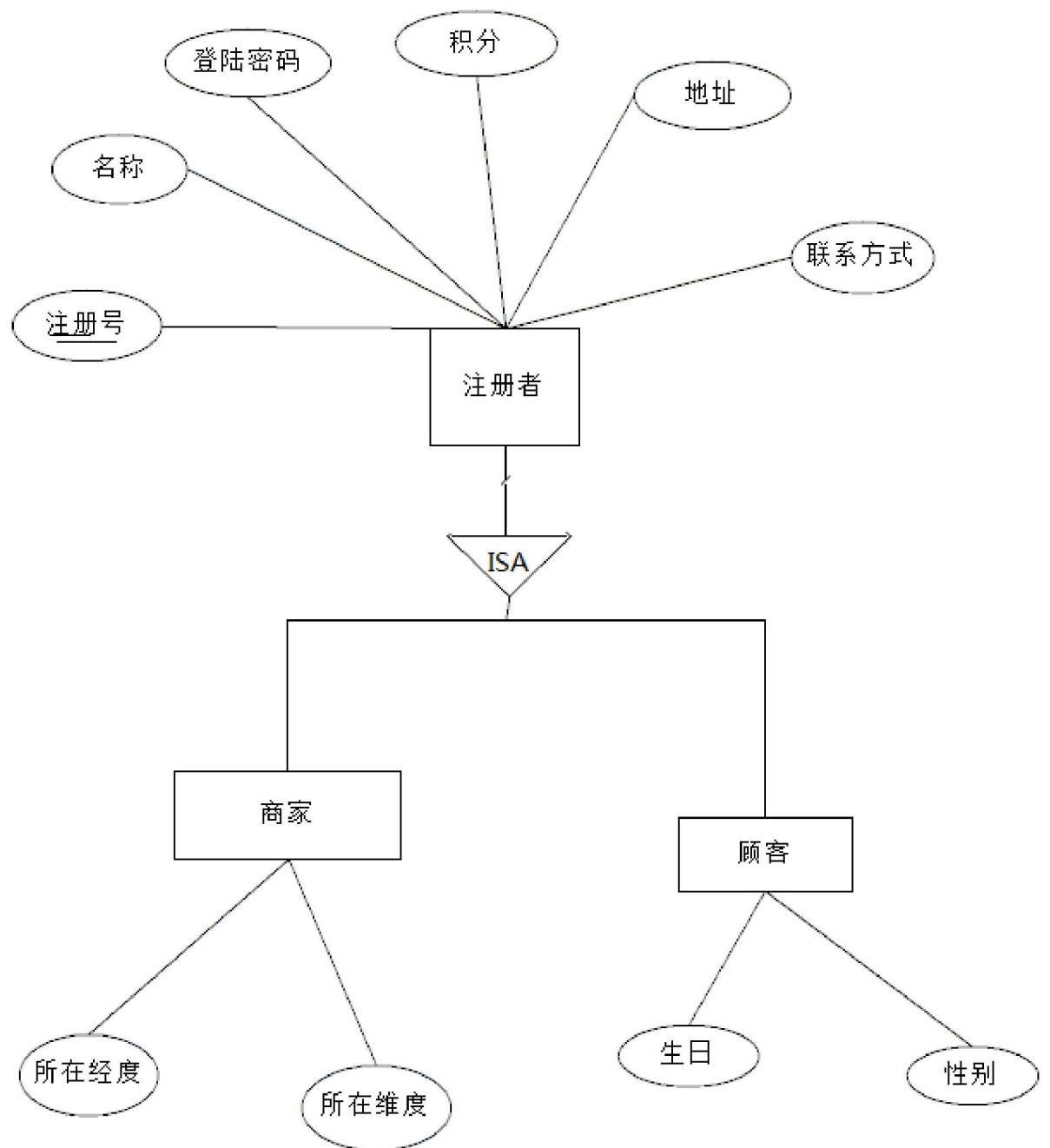
# 3 结构设计

## 3.1 概念结构设计

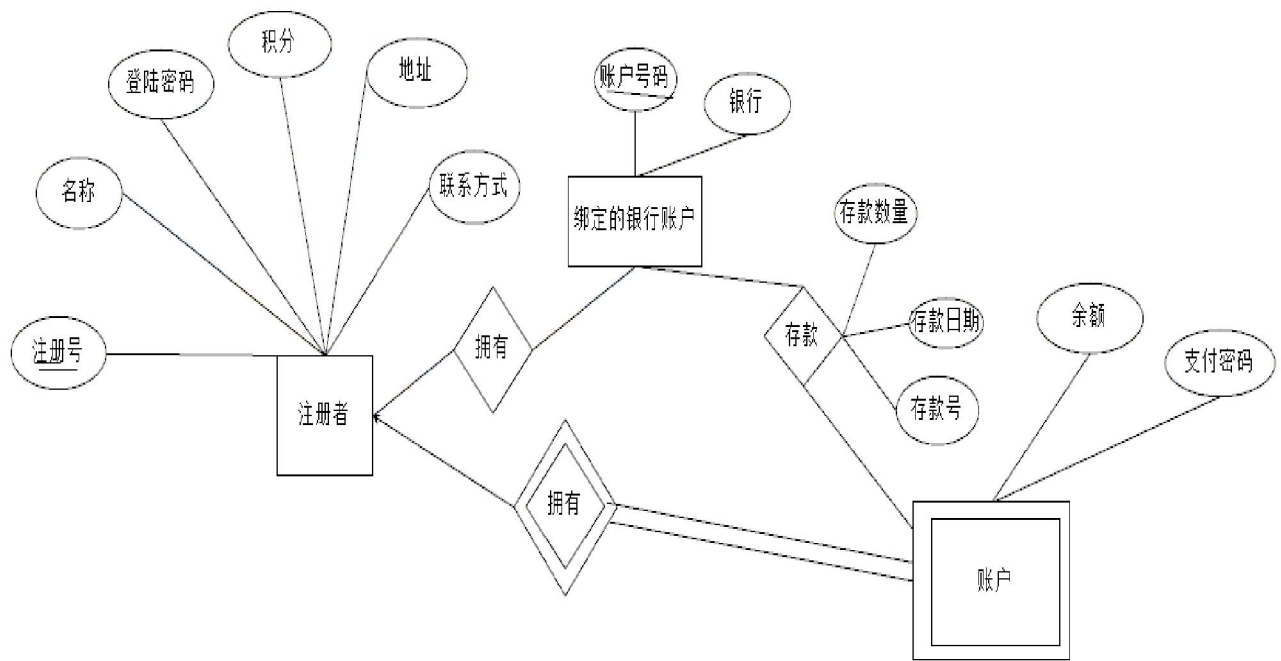
### 3.1.1 局部视图

注册者基本信息:

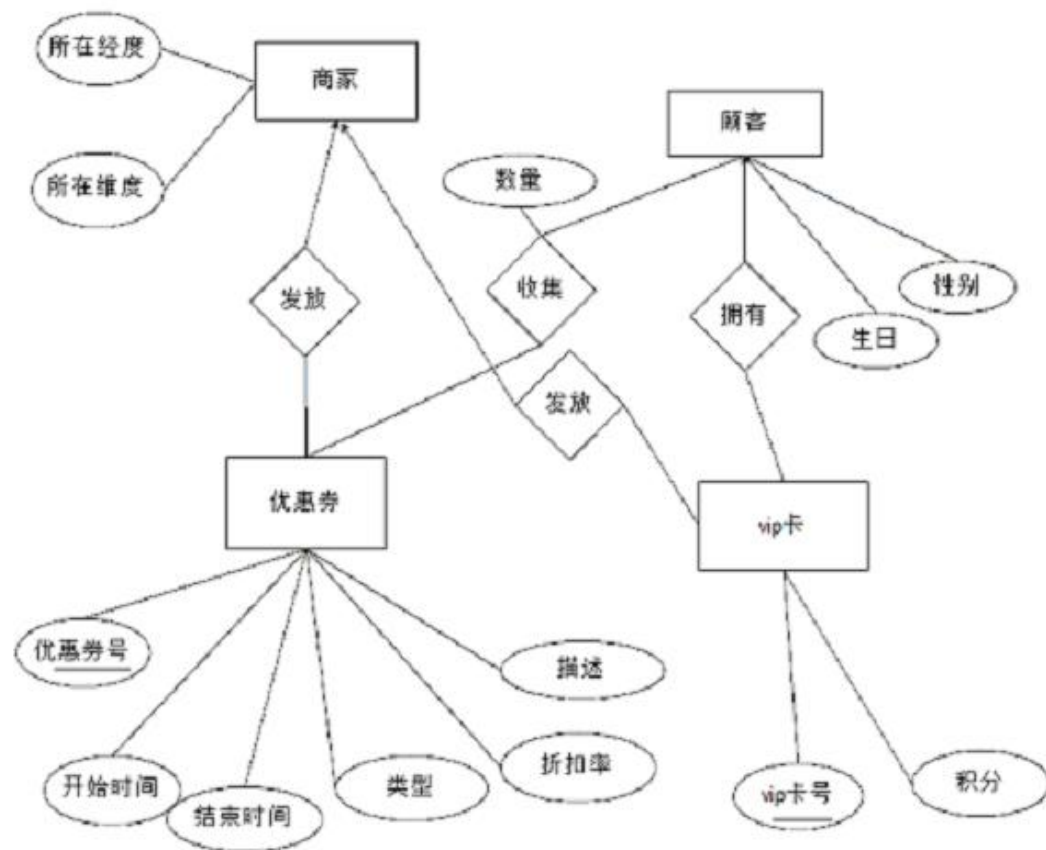
注册者基本信息:



注册者账户管理:



优惠券、VIP 卡管理：

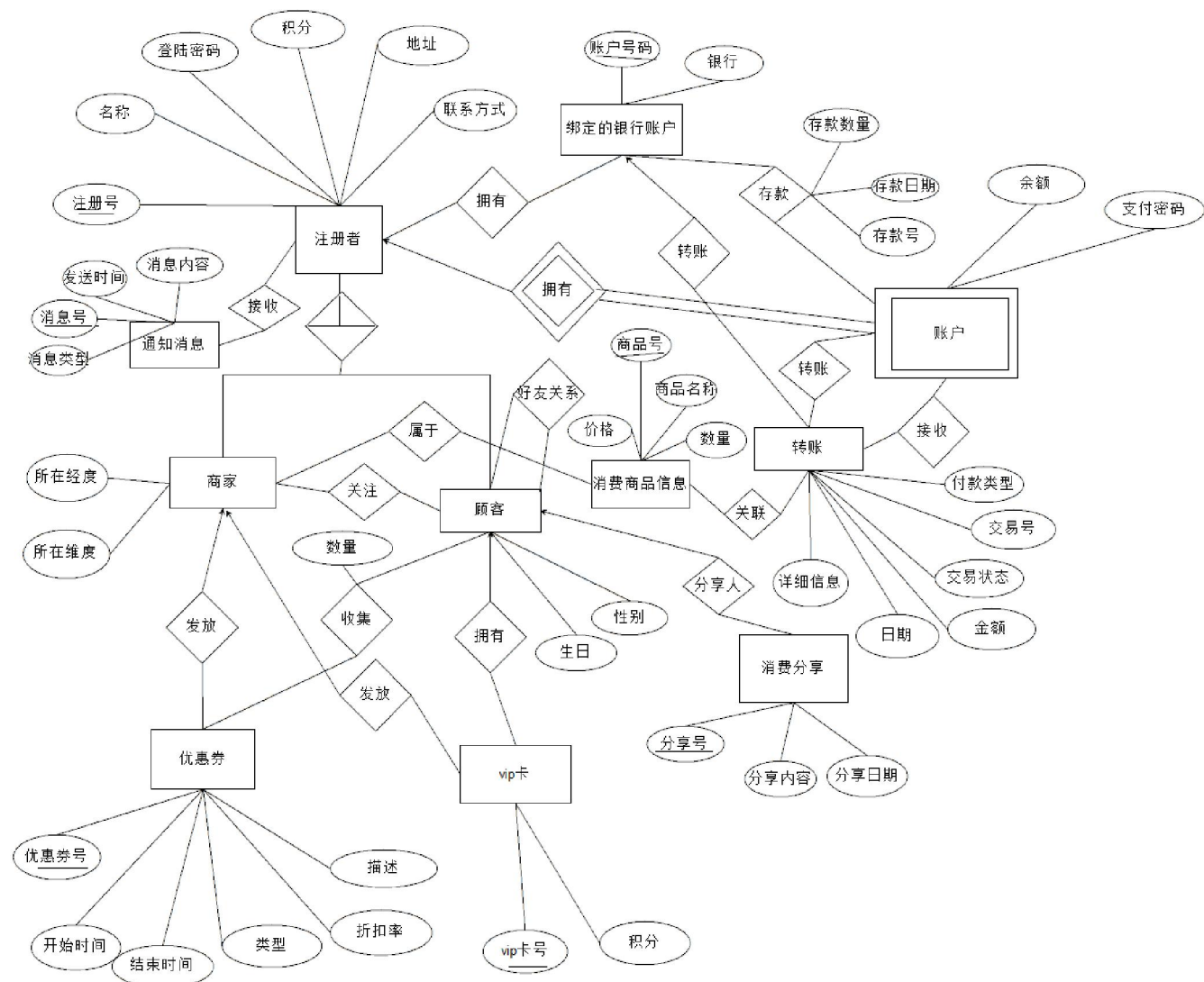


转账记录管理:





### 3.1.2 综合视图



### 3.2 逻辑结构设计

关系模式:

注册者 (注册号, 名称, 邮箱, 电话号码, 地址, 登录密码, 积分)

商家 (注册号, 所在经度, 所在纬度)

客户 (注册号, 性别, 生日)

账户（账户号，支付密码，余额）  
 绑定银行卡（注册号，银行账户号码，银行名称）  
 存款记录（存款号，账户号，银行账户号，存款金额，存款日期）  
 优惠券（优惠券号，发放商家，开始时间，结束时间，优惠类型，折扣程度，描述）  
 优惠券收藏（优惠券号，注册号，收藏数量）  
 VIP 卡（卡号，发放商家，拥有者，积分）  
 转账记录（转账号，转账人注册号，接收注册号，转账金额，转账时间，转账类型，  
 详细信息）  
 转账银行账户（转账号，银行账户号）  
 消费商品信息（商品号，关联转账号，商品名称，商品价格，数量）  
 消费分享（分享号，分享人注册号，分享内容，分享日期）  
 通知消息（消息号，接收人注册号，消息内容，消息类型）  
 好友关系（客户 1，客户 2）  
 关注商家（客户注册号，商家注册号）

视图：

客户基本信息 (注册号，名称，邮箱，手机号码，地址，积分，性别，生日)  
 客户重要信息 (注册号，登陆密码，支付密码，余额)  
 商家基本信息 (注册号，名称，邮箱，手机号码，地址，积分，所在经度，所在纬度)  
 商家重要信息 (注册号，登陆密码，支付密码，余额)  
 客户优惠券收集 (册号，优惠券号，数量，开始时间，结束时间，优惠类型，折扣程度)

SQL 语句：

```

use coupon;
create table register (
  registerId int auto_increment primary key,
  name varchar(30) not null,
  loginPasswor char(128) not null,
  email char(40),
  phoneNumber varchar(40),
  address varchar(40),
  grade int not null
);

create table customer (
  customerId int,
  sex char(1) check (sex in ('男', '女')),
  birthday date,
  foreign key(customerId) references register(registerId)
);

create table merchant (
  merchantId int,

```

```

        longitude decimal(9 , 6 ),
        latitude decimal(9 , 6 ),
        foreign key(merchantId) references register(registerId)
    );

```

```

create table account (
    accountId int primary key,
    foreign key (accountId)
        references register (registerId),
    balance double (10 , 2 ) check (balance >= 0),
    payPassword char(128) not null
);

```

```

create table bindedBankAccount (
    bankAccountId varchar(25) primary key,
    registerId int,
    foreign key (registerId)
        references register (registerId),
    bank varchar(30) not null
);

```

```

create table depositRecord (
    depositId int auto_increment primary key,
    bankAccountId varchar(25),
    amount double (10 , 2 ) check (amount > 0),
    depositDate date not null,
    foreign key (bankAccountId)
        references bindedBankAccount (bankAccountId)
);

```

```

create table coupon (
    couponId int auto_increment primary key,
    merchantId int,
    startDate date,
    endDate date not null,
    couponType varchar(10) not null,
    discountRate double (10 , 2 ),
    decription varchar(200),
    foreign key (merchantId)
        references merchant (merchantId)
);

```

```

create table couponCollection (

```

```

        customerId int,
        couponId int,
        number int,
        primary key (customerId , couponId),
        foreign key (customerId)
            references customer (customerId),
        foreign key (couponId)
            references coupon (couponId)
    );

create table vipCard (
    cardId varchar(25) primary key,
    merchantId int,
    customerId int,
    grade int,
    foreign key (merchantId)
        references merchant (merchantId),
    foreign key (customerId)
        references customer (customerId)
);

create table transactionRecord (
    transactionId char(22) primary key,
    sender int,
    receiver int,
    amount double (10 , 2 ) not null,
    transactionDate datetime not null,
    transactionType varchar(15) not null,
    detail text,
    foreign key (sender)
        references register (registerId),
    foreign key (receiver)
        references register (registerId)
);

create table productInformation (
    productId varchar(20),
    transactionId char(22),
    primary key (productId , transactionId),
    foreign key (transactionId)
        references transactionRecord (transactionId)
);

create table sharing (

```

```

        sharingId int auto_increment primary key,
        shareId int,
        content varchar(140) not null,
        sharingDate datetime not null,
        foreign key (shareId)
            references customer (customerId)
    );

create table message (
    messageId int auto_increment primary key,
    receiverId int,
    content varchar(200) not null,
    messageType varchar(15) not null,
    foreign key (receiverId)
        references register (registerId)
);

create table friendship (
    customerId1 int,
    customerId2 int,
    primary key (customerId1 , customerId2),
    foreign key (customerId1)
        references customer (customerId),
    foreign key (customerId2)
        references customer (customerId)
);

create table followdMerchant (
    merchantId int,
    customerId int,
    primary key (merchantId , customerId),
    foreign key (merchantId)
        references merchant (merchantId),
    foreign key (customerId)
        references customer (customerId)
);

```

### 3.3 物理结构设计

创建索引：各表的主键自动生成索引。另外，根据查询特点，对转账记录表的转账时间建立索引，对消费分享的分享日期建立索引, SQL 语句如下：

create index transactionDateIndex on transactionRecord(transactionDate desc);

create index sharingDateIndex on sharing(sharingDate desc)

## 4 运用设计

### 4.1 数据字典设计

注册者表:

数据项名称	数据项标识符	数据类型	约束	备注
注册号	registerId	自增长 int	主键	用于标识不同的注册者
名称	name	varchar (30)	非空	
邮箱	email	varchar(40)	非空	注册邮箱
电话号码	phoneNumber	varchar(20)		
登陆密码	loginPassword	char(128)	非空	Md5 加密
积分	grade	int	非空	
地址	address	varchar(40)		

商家表:

数据项名称	数据项标识符	数据类型	约束	备注
注册号	merchantId	Int	外键	引用注册者表主键
所在经度	longitude	decimal(9,6)		精度要求小数点后六位
所在纬度	latitude	decimal(9,6)		精度要求小数点后六位

客户表:

数据项名称	数据项标识符	数据类型	约束	备注
注册号	customerId	Int	外键	引用注册者表主键
性别	sex	char(1)	check in ('男','女')	
生日	birthday	Date		

账户表:

数据项名称	数据项标识符	数据类型	约束	备注
注册号	accountId	Int	外键	一个注册号对应

				一个账户，用注册号做主键
余额	balance	double(12,2)	非空	精确到小数点后两位
支付密码	payPassword	char(128)	非空	Md5 加密

绑定银行账户表：

数据项名称	数据项标识符	数据类型	约束	备注
注册号	registered	int	外键，主键	外键引用注册者表，与银行账户号构成联合主键
银行账户号	bankAccountId	varchar(25)	主键	与注册号构成联合主键
银行名称	bank	varchar(30)	非空	

存款记录表：

数据项名称	数据项标识符	数据类型	约束	备注
存款号	depositId	自增长 int	主键	用于标识不同的存款
账户号	accountId	int	外键	引用账户表
银行账户号	bankAccountId	varchar(25)	外键	引用绑定银行账户表
存款金额	amount		check(amount > 0)	精确到小数点后两位，大于零
存款时间	depositDate	date	非空	

优惠券表：

数据项名称	数据项标识符	数据类型	约束	备注
优惠券号	couponId	自增长 int	主键	标识不同的优惠券
发放商家	merchantId	Int	外键	引用商家表
开始时间	startDate	date		
结束时间	endDate	date	非空	优惠券到期期限
优惠类型	couponType	varchar(10)	非空	
折扣程度	discountRate	double(10,2)	非空	
描述	description	varchar (200)		对消费券的简要说明

优惠券收藏表:

数据项名称	数据项标识符	数据类型	约束	备注
优惠券号	couponed	int	外键	引用优惠券表
注册号	customerId	int	外键	引用客户表
收藏数量	number	int	非空	

VIP 卡表:

数据项名称	数据项标识符	数据类型	约束	备注
卡号	cardId	varchar(25)	主键	
发放商家	merchantId	int	外键	引用商家表
拥有者	owner	int	外键	引用客户表
积分	grade	int	非空	

转账记录表:

数据项名称	数据项标识符	数据类型	约束	备注
转账号	transactionId	char(22)	主键	有系统自动生成
转账人注册号	sender	int	外键	引用账户表
接收人注册号	receiver	int	外键	引用账户表
转账金额	amount	double(10,2)	非空	精确到小数点后两位
转账时间	transactionDate	datetime	非空	
转账类型	transactionType	varchar(15)	非空	
详细信息	detail	text		对本次转账的描述

转账银行账户 (转账号, 银行账户号)

数据项名称	数据项标识符	数据类型	约束	备注
转账号	transactionId	char(22)	主键,外键	引用转账记录表
银行账户号	bankAccountId	varchar(25)	外键	引用绑定银行账户表

数据项名称	数据项标识符	数据类型	约束	备注
商品号	productId	varchar(20)	主键	
对应转账号	transactionId	char(22)	主键, 外键	引用转账记录表
商品名称	productName	varchar(25)	非空	
商品价格	price	double(10,2)	非空	
数量	quantity	double(10,2)	非空	



消费分享表:

数据项名称	数据项标识符	数据类型	约束	备注
分享号	sharingId	自增长 int	主键	
分享人注册号	sharerId	int	外键	引用客户表
分享内容	content	varchar(140)	非空	
分享日期	sharingDate	datetime	非空	

通知消息表:

数据项名称	数据项标识符	数据类型	约束	备注
消息号	messageId	自增长 int	主键	
接收人注册号	receiverId	int	外键	引用注册者表
消息内容	content	varchar(200)	非空	
消息类型	messageType	varchar(15)	非空	

好友关系表:

数据项名称	数据项标识符	数据类型	约束	备注
客户 1	customerId1	int	外键, 主键	引用客户表, 与 customerId2 构成联合主键
客户 2	customerId2	int	外键	引用客户表, 与 customerId1 构成联合主键

关注商家表:

数据项名称	数据项标识符	数据类型	约束	备注
客户注册号	customerId	int	外键, 主键	引用客户表, 与 sellerId 构成联合主键
商家注册号	merchantId	int	外键, 主键	引用商家表, 与 customerId 构成联合主键

## 4.2 安全保密设计

一般访问者访问数据都通过程序进行, 用程序来控制不同用户访问的权限, 用户只有输

入口令，通过了权限认证才能调用相应的函数来操作数据，程序跟踪检测访问者的增删改查操作。

为不同的系统管理者建立不同的视图，赋予相应的权限来控制系统管理者对数据的访问，访问数据前操作员也须通过口令验证。