**“银行业务管理系统”**

系统设计与实现报告

**姓名：雷雨轩**

**学号：PB18111791**

**计算机科学与技术学院**

**中国科学技术大学**

2021年7月

**目 录**

1 概述 1

1.1 系统目标 1

1.2 需求说明 1

1.3 本报告的主要贡献 1

2 总体设计 1

2.1 系统模块结构 1

2.2 系统工作流程 1

2.3 数据库设计 1

3 详细设计【可选】 2

3.1 \*\*\*\* 模块 2

3.2 \*\*\*\* 模块 2

3.3 \*\*\*\* 模块 2

4 实现与测试 2

4.1 实现结果 2

4.2 测试结果 2

4.3 实现中的难点问题及解决【可选】 2

5 总结与讨论 3

# 

# 1 概述

## 系统目标

## 开发一个银行业务管理系统，实现相关功能。

## 需求说明

## 银行系统数据需求：

## 银行有多个支行。各个支行位于某个城市，每个支行有唯一的名字。银行要监控每个支行的资产。 银行的客户通过其身份证号来标识。银行存储每个客户的姓名、联系电话以及家庭住址。 为了安全起见，银行还要求客户提供一位联系人的信息，包括联系人姓名、手机号、 Email 以及与客户的关系。 客户可以有帐户，并且可以贷款。客户可能和某个银行员工发生联系，该员工是此客户的贷款负责人或银行帐户负责人。 银行员工也通过身份证号来标识。员工分为部门经理和普通员工，每个部门经理都负责领导其所在部门的员工，并且每个员工只允许在一个部门内工作。每个支行的管理机构存储每个员工的姓名、电话号码、家庭地址、所在的部门号、部门名称、部门类型及部门经理的身份证号。银行还需知道每个员工开始工作的日期，由此日期可以推知员工的雇佣期。 银行提供两类帐户——储蓄帐户和支票帐户。帐户可以由多个客户所共有，一个客户也可开设多个账户，但在一个支行内最多只能开设一个储蓄账户和一个支票账户。每个帐户被赋以唯一的帐户号。银行记录每个帐户的余额、 开户日期、 开户的支行名以及每个帐户所有者访问该帐户的最近日期。另外，每个储蓄帐户有利率和货币类型，且每个支票帐户有透支额。 每笔贷款由某个分支机构发放，能被一个或多个客户所共有。每笔贷款用唯一的贷款号标识。银行需要知道每笔贷款所贷金额以及逐次支付的情况（银行将贷款分几次付给客户）。虽然贷款号不能唯一标识银行所有为贷款所付的款项，但可以唯一标识为某贷款所付的款项。对每次的付款需要记录日期和金额

## 银行系统功能需求：

## 客户管理： 提供客户所有信息的增、删、改、查功能； 如果客户存在着关联账户或者贷款记录，则不允许删除；

## 账户管理： 提供账户开户、销户、修改、查询功能，包括储蓄账户和支票账户； 账户号不允许修改；

## 贷款管理： 提供贷款信息的增、删、查功能，提供贷款发放功能； 贷款信息一旦添加成功后不允许修改； 要求能查询每笔贷款的当前状态（未开始发放、发放中、已全部发放） ； 处于发放中状态的贷款记录不允许删除；

## 业务统计： 按业务分类（储蓄、 贷款）和时间（月、 季、 年）统计各个支行的业务总金额和用户数， 统计的结果以表格形式展示

注释：

* 支行、部门和员工的信息需要预先插入到数据库中，本项目假设这三类数据已经在数据库中了，并且本实验不要求实现这三类数据的维护
* 查询功能允许自行设计，但要求尽可能灵活设计，考虑用户多样化的查询需求
* 各类数据的类型可自行根据实际情况设计
* 测试数据自行设计

## 1.3 本报告的主要贡献

对基于MySQL、Flask、SQLAlchemy的B/S架构的数据库系统的开发流程做了详细介绍，并且做了结果展示与细节分析、总结。

# 2 总体设计

## 2.1 系统模块结构

database

demo.sql -- 数据库表项建立

data.sql --数据库数据初始化

static -- 文件夹下包含实现前端界面及交互的css、icons、js的模块库代码

css

icons

js

templates

account --账户管理

create.html --创建账户

search.html --查询、删除、更新账户信息

customer --客户管理

create.html --创建客户

search.html --查询、删除、更新客户信息

loan --贷款管理

create.html --创建贷款

search.html --查询、发放、删除贷款

error.html -- page not found

index.html -- 银行管理系统主界面

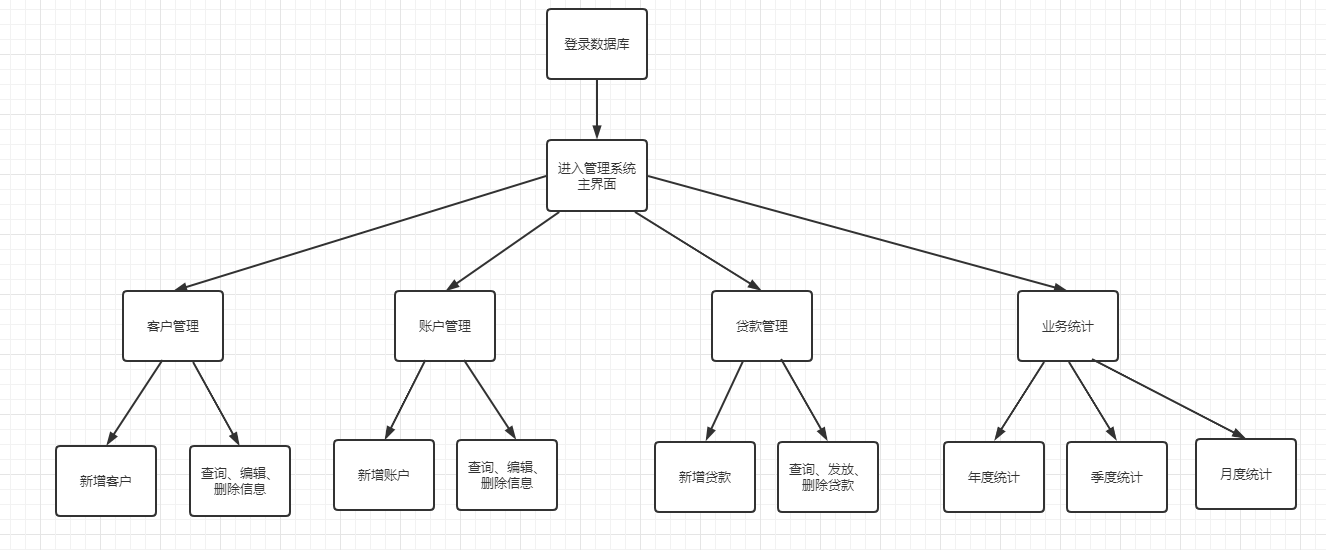
statistics.html --统计年、季度、月份信息

config.py --数据库连接的账户设置

models.py --SQLAlchemy操作所需数据库表项信息

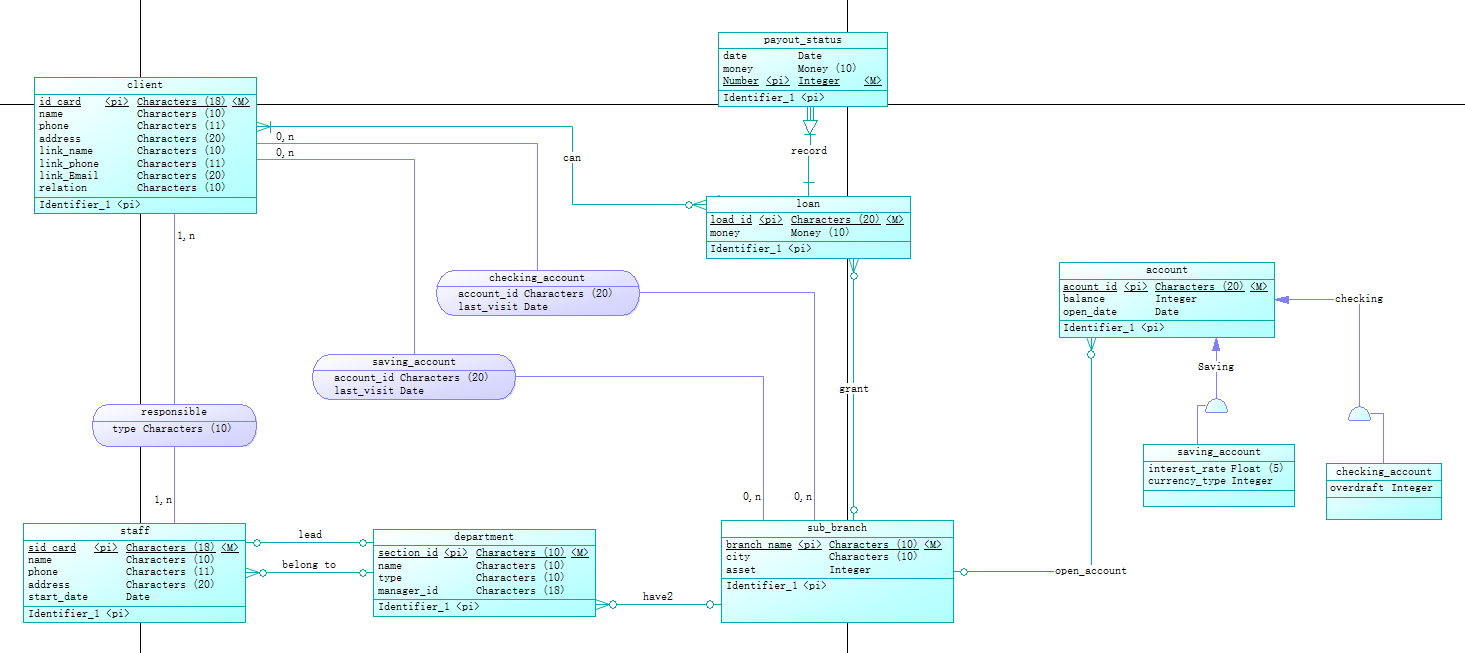
my\_bank.py --运行文件，包括各种路由函数

## 2.2 系统工作流程



## 2.3 数据库设计

由Lab2里用Powerdesigner生成的ER图



详细的物理数据库设计即参考database文件夹下的demo.sql文件，可能各部分具体的名称有一些变化。

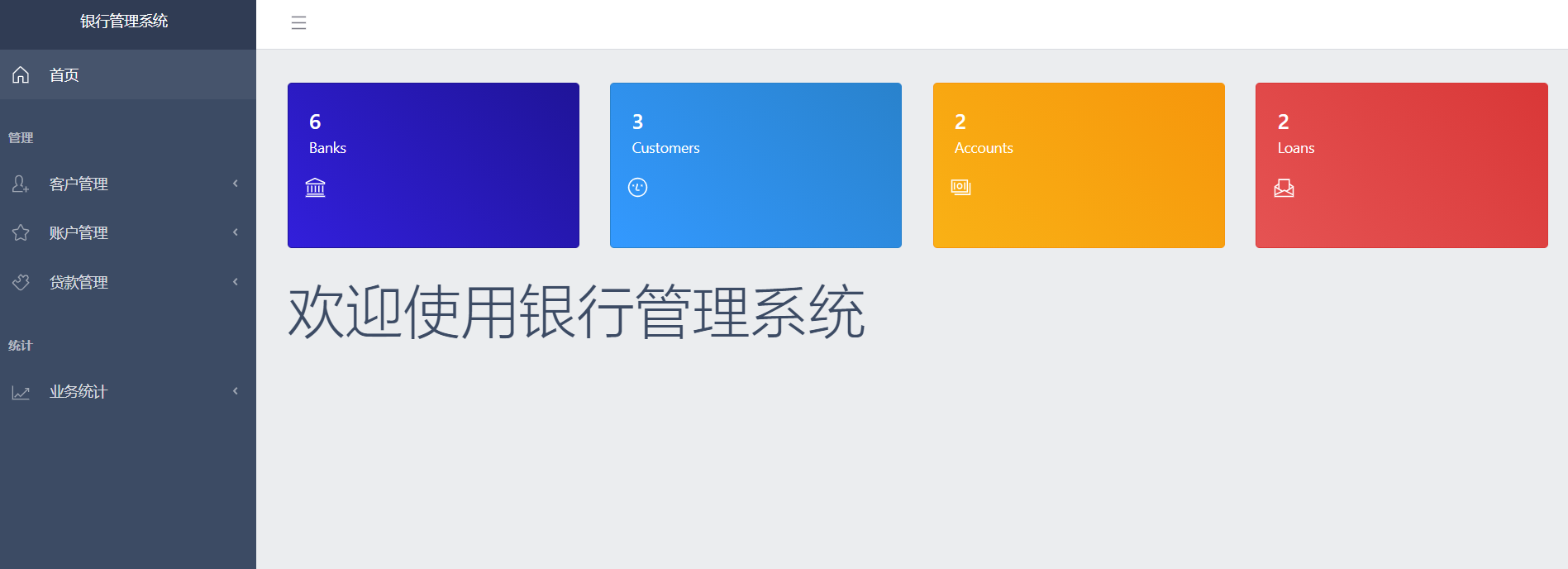
# 3 详细设计【可选】

# 4 实现与测试

## 4.1 实现结果

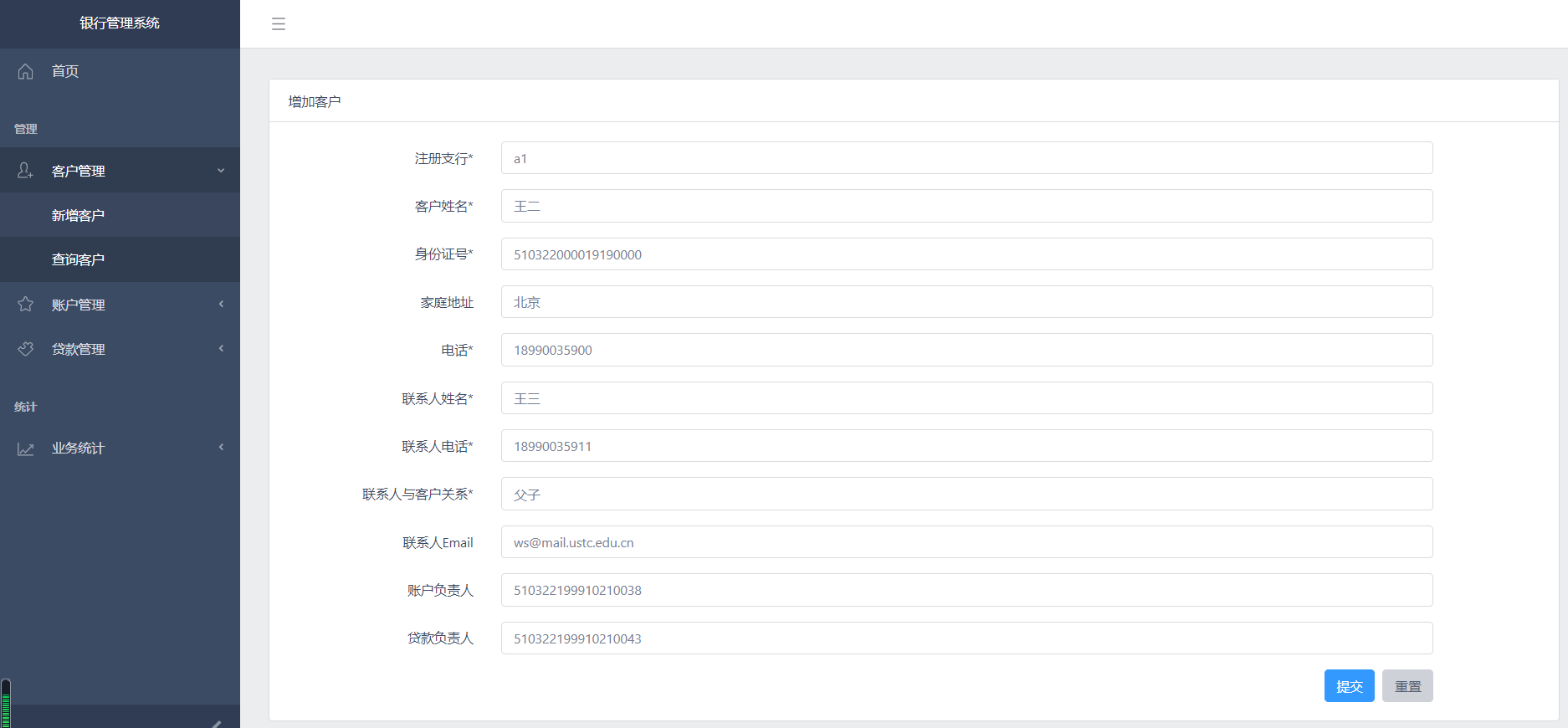
// 给出各个功能需求的实现界面和运行结果。

**主界面**

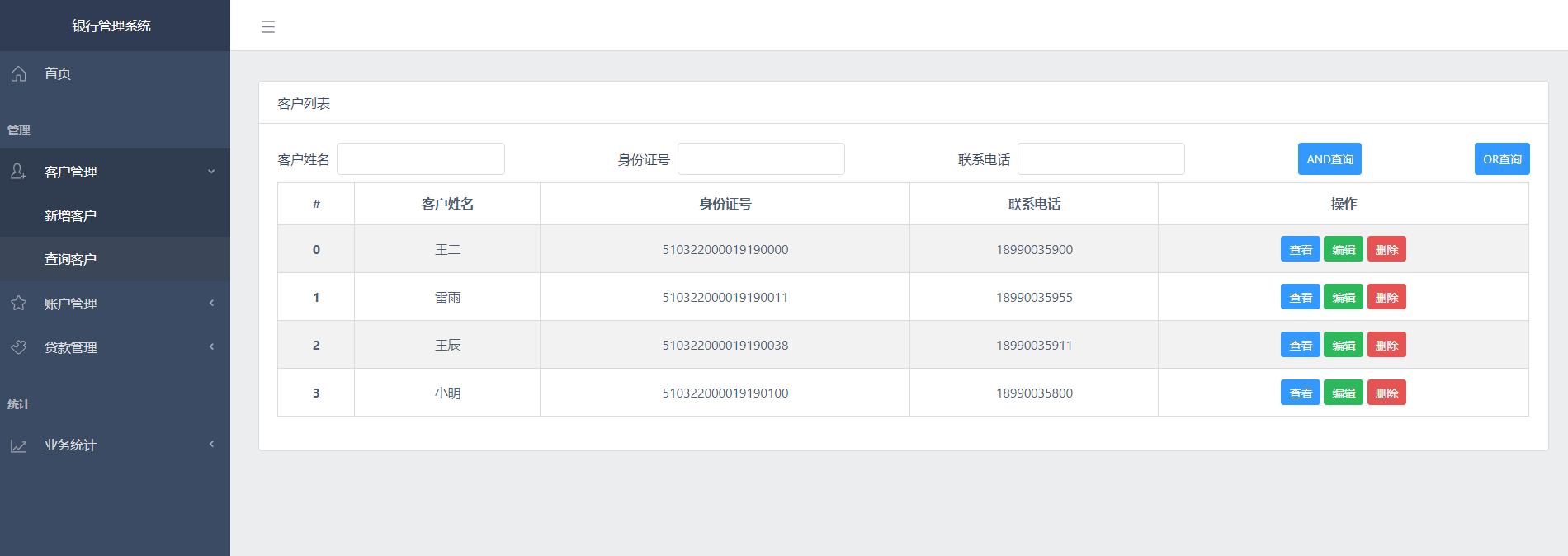


**客户管理**

**新增客户**



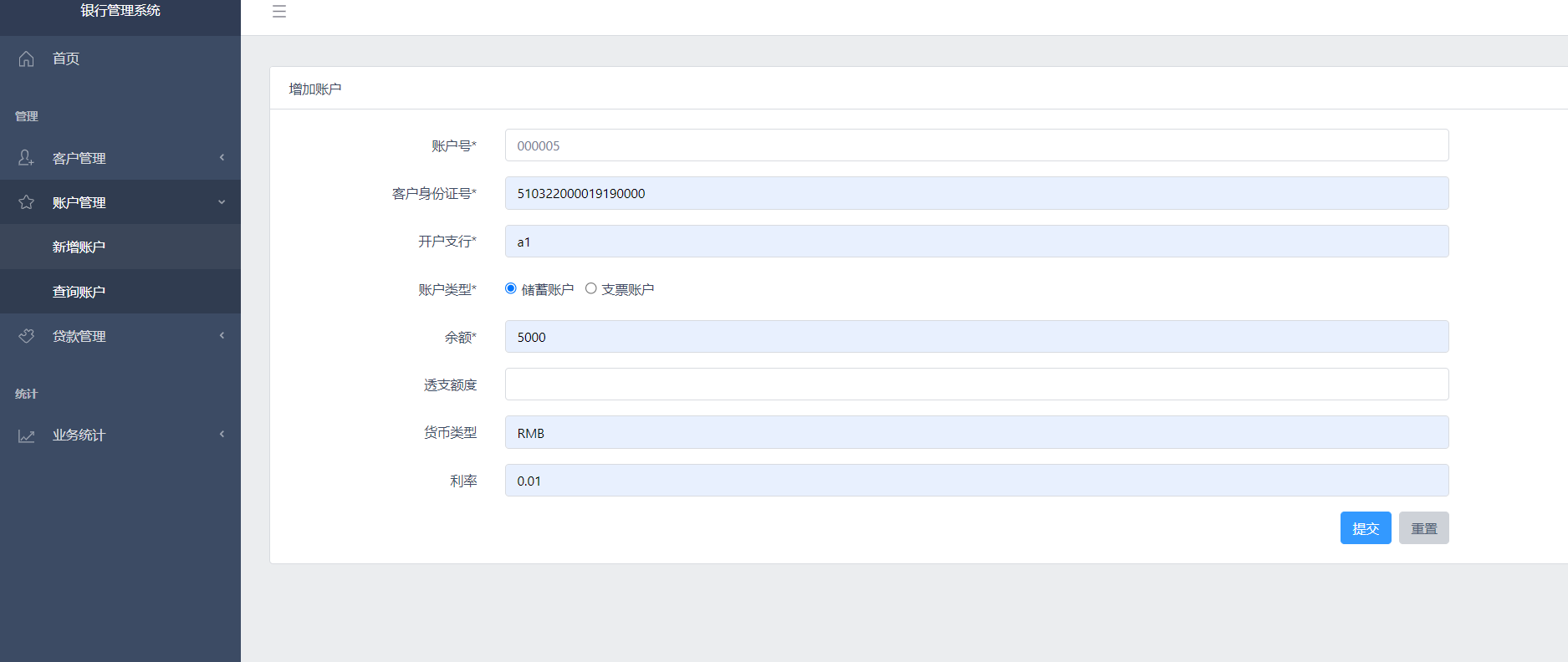
**查询客户**



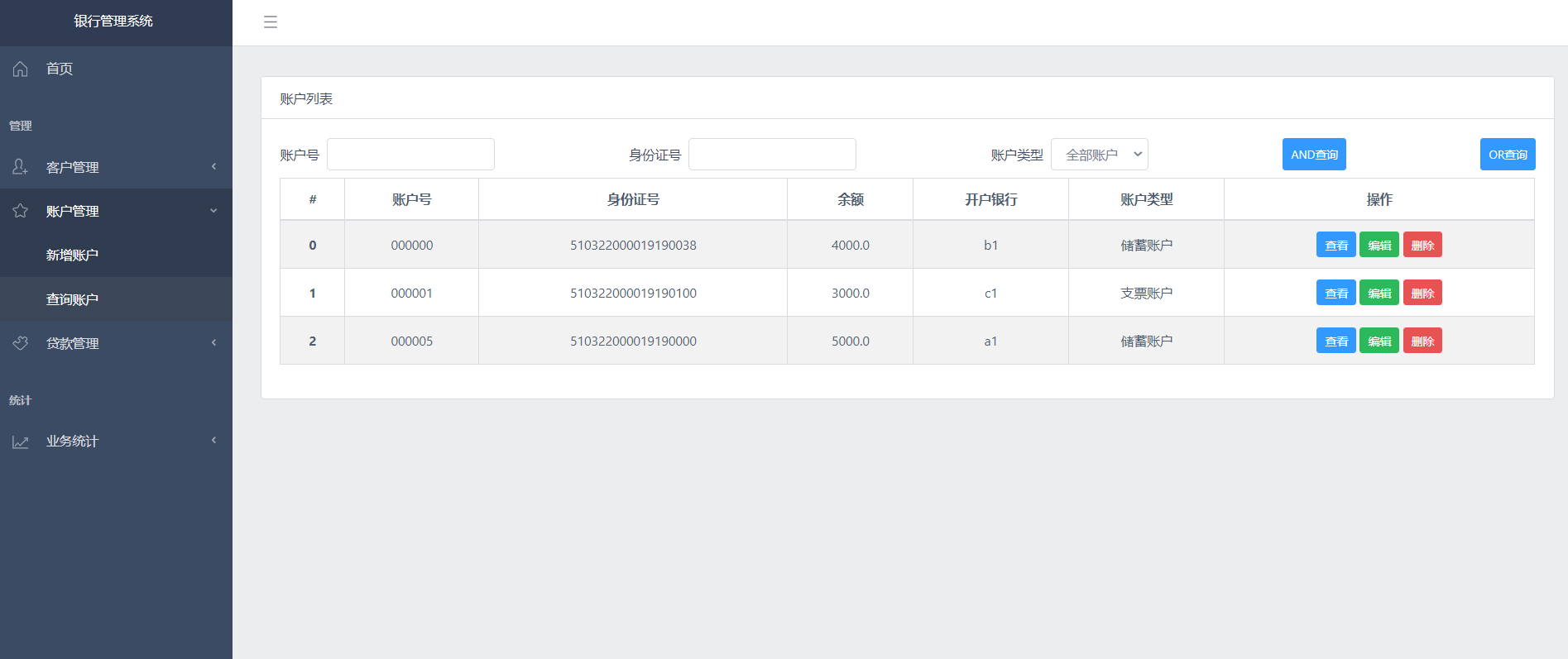
可以发现前面新加的王二已经在里面

**账户管理**

**新增账户**



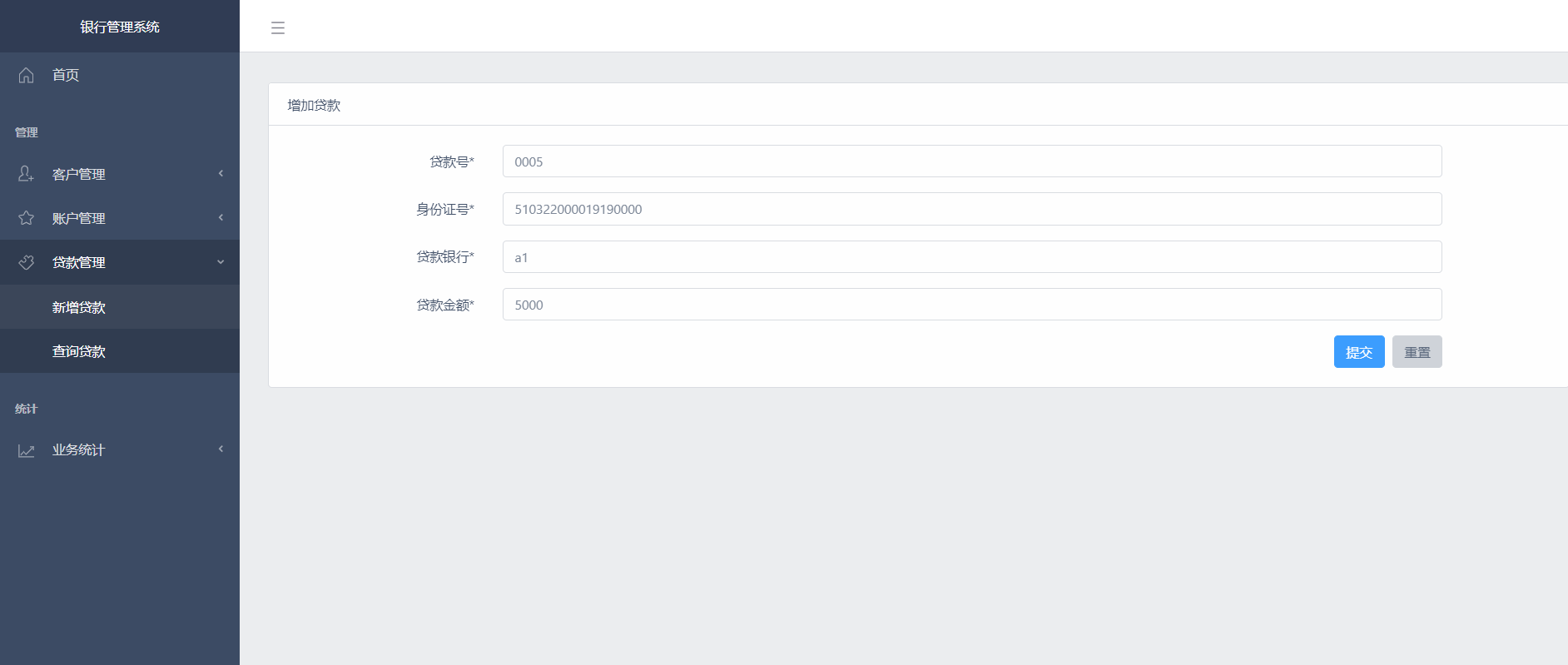
**查询账户**



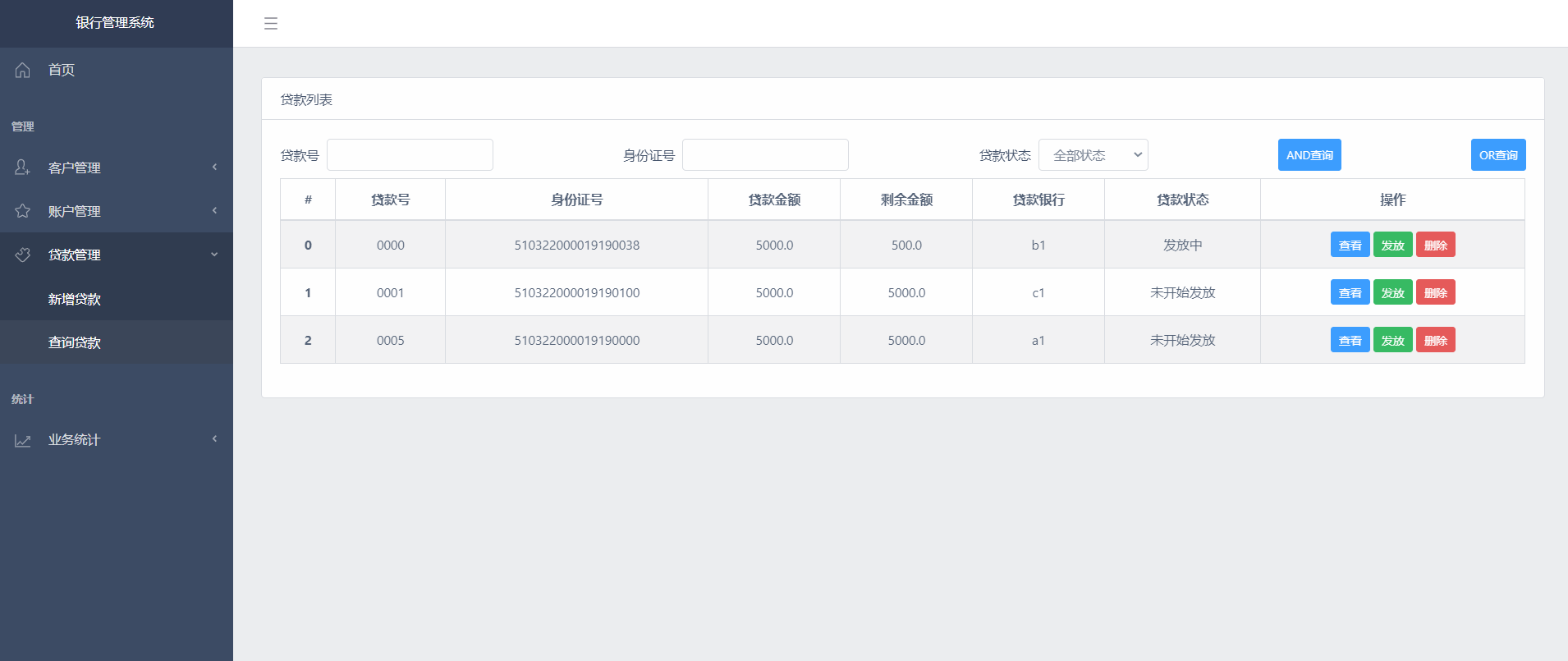
可以看到前面新加的账户000005已经显示

**贷款管理**

**新增贷款**



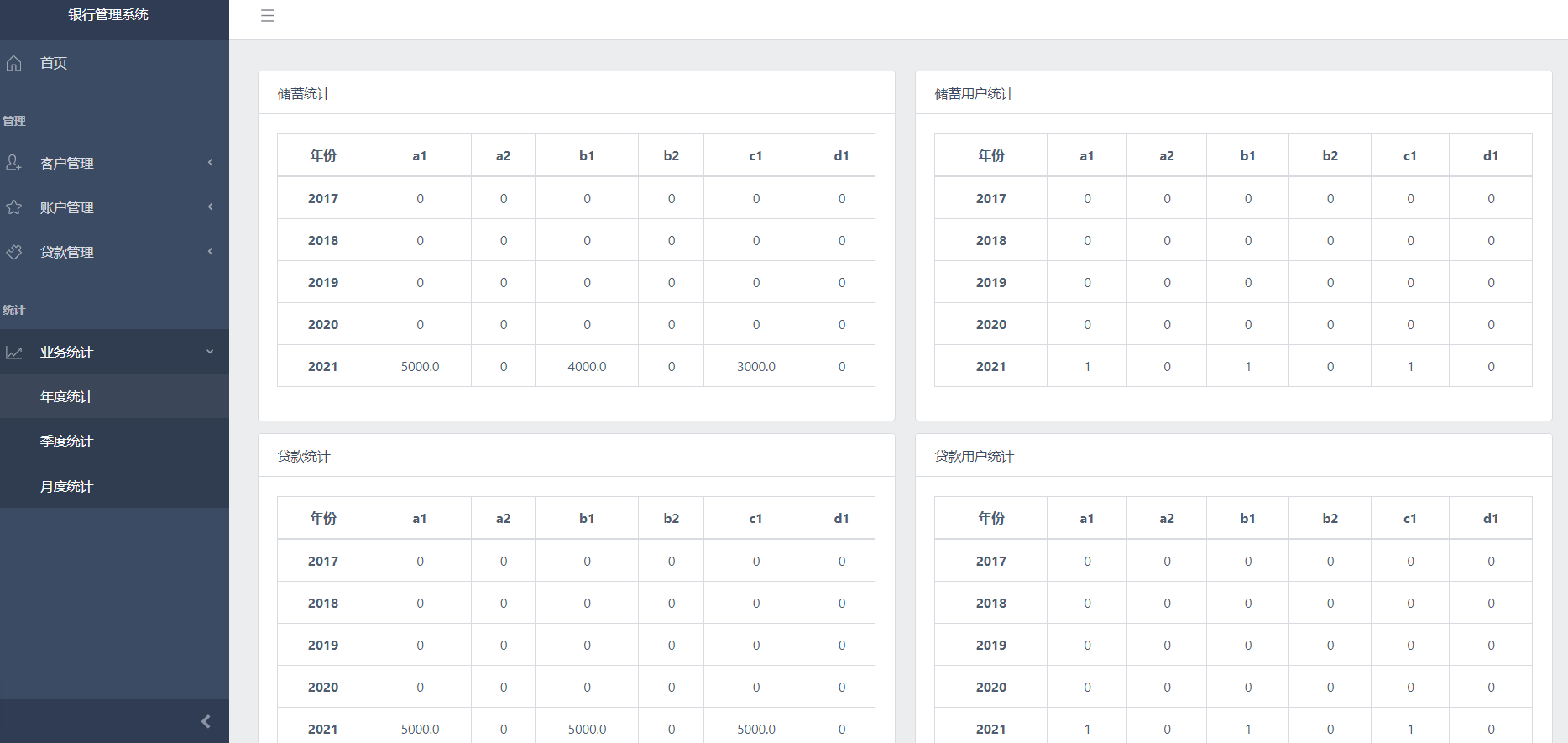
**查询贷款**



可以看到前面新加的贷款0005已在里面

**业务统计**

**年度统计**



图中每个表的列为年份/季度/月份，行为对应的支行。储蓄、贷款统计的是金额，用户统计则指用户创建数目。

季度、月度统计类似

## 4.2 测试结果

增加各类对象的功能已经在4.1里展示过了，现在主要展示每个模块的其他功能

**客户管理**

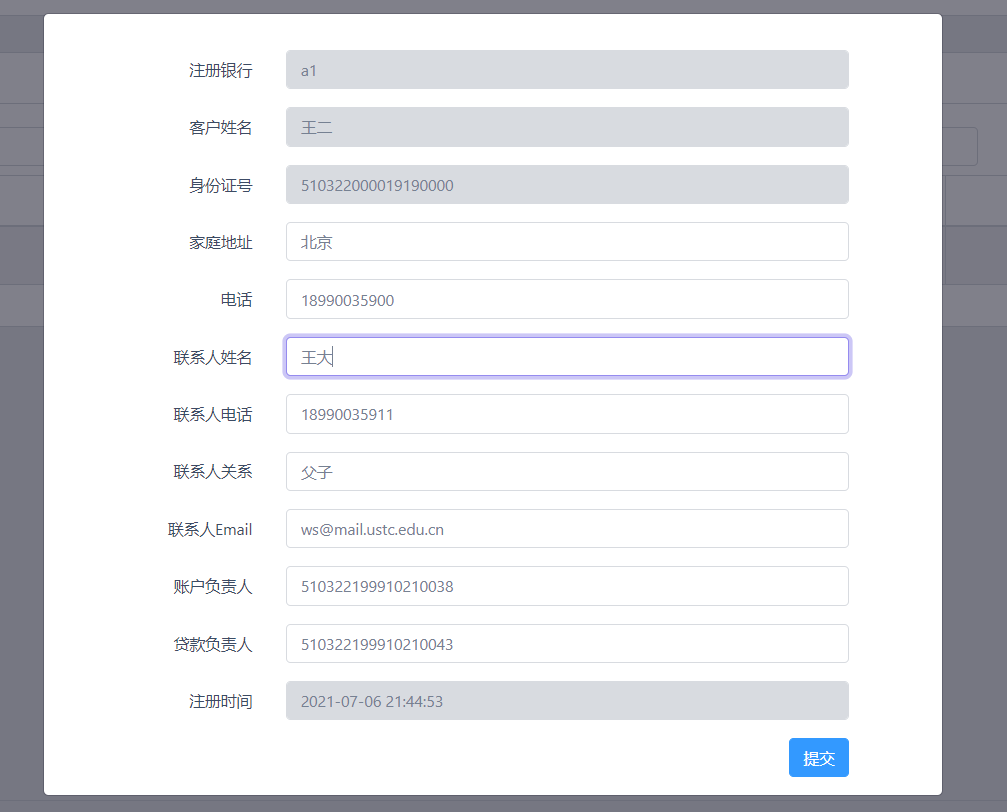
提供AND和OR的查询，比如AND查询如下，查询姓名为王二，身份证为510322000019190000的人。



点击查看可以看到详细信息



点击编辑可以实现数据修改，比如下面把联系人姓名改为王大

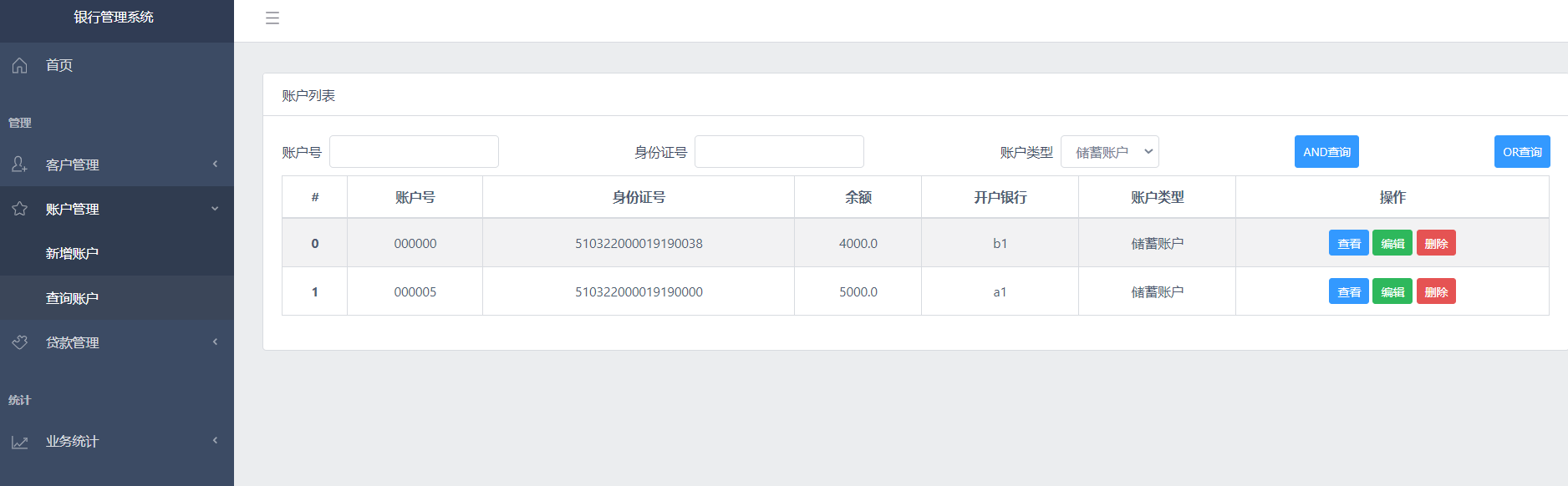


点击删除则能删除对应的用户，当然如果有关联账户或者贷款则无法删除，显示删除失败



**账户管理**

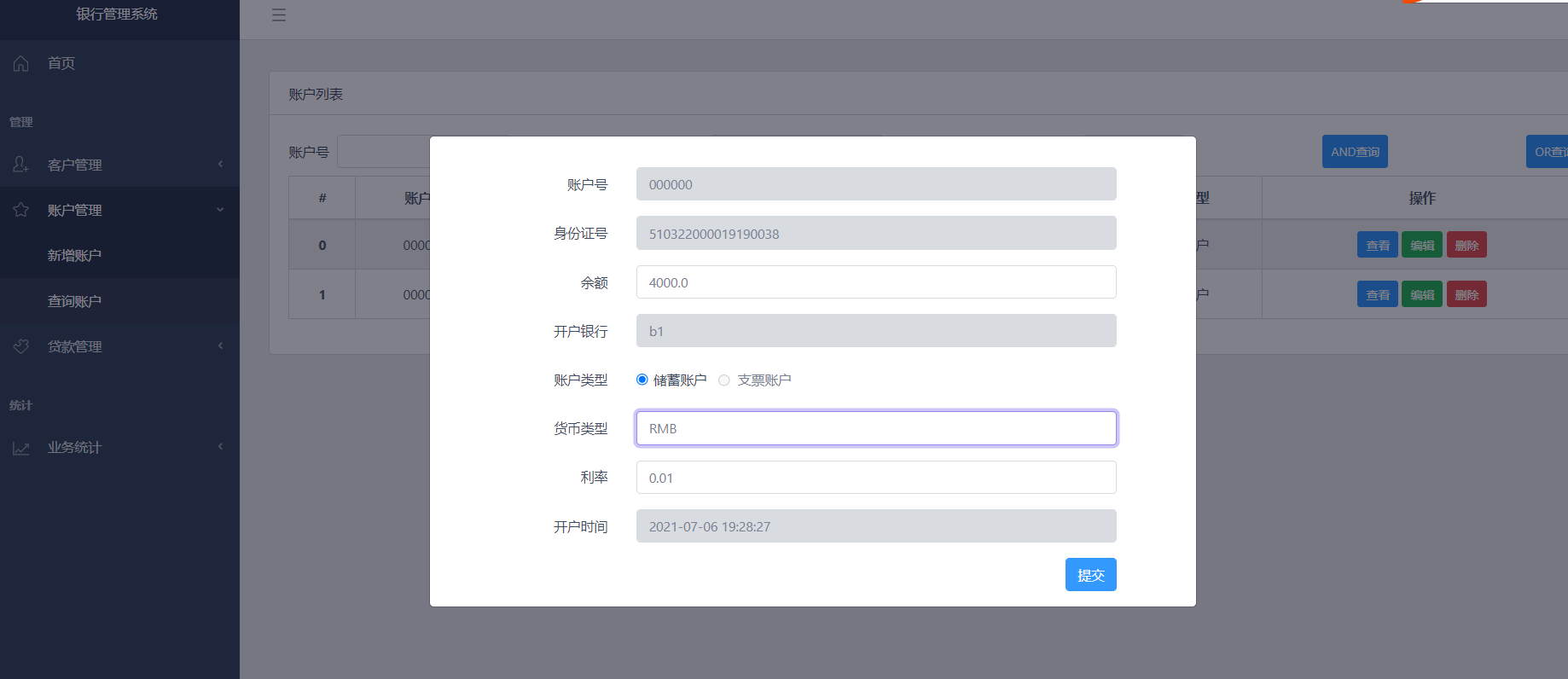
同样提供AND和OR查询，这里以OR查询为例，查询所有储蓄账户



点击查看可以看到对应账户详细信息



点击编辑则可以进行某些信息的修改



点击删除则可以删除对应账户



**贷款管理**

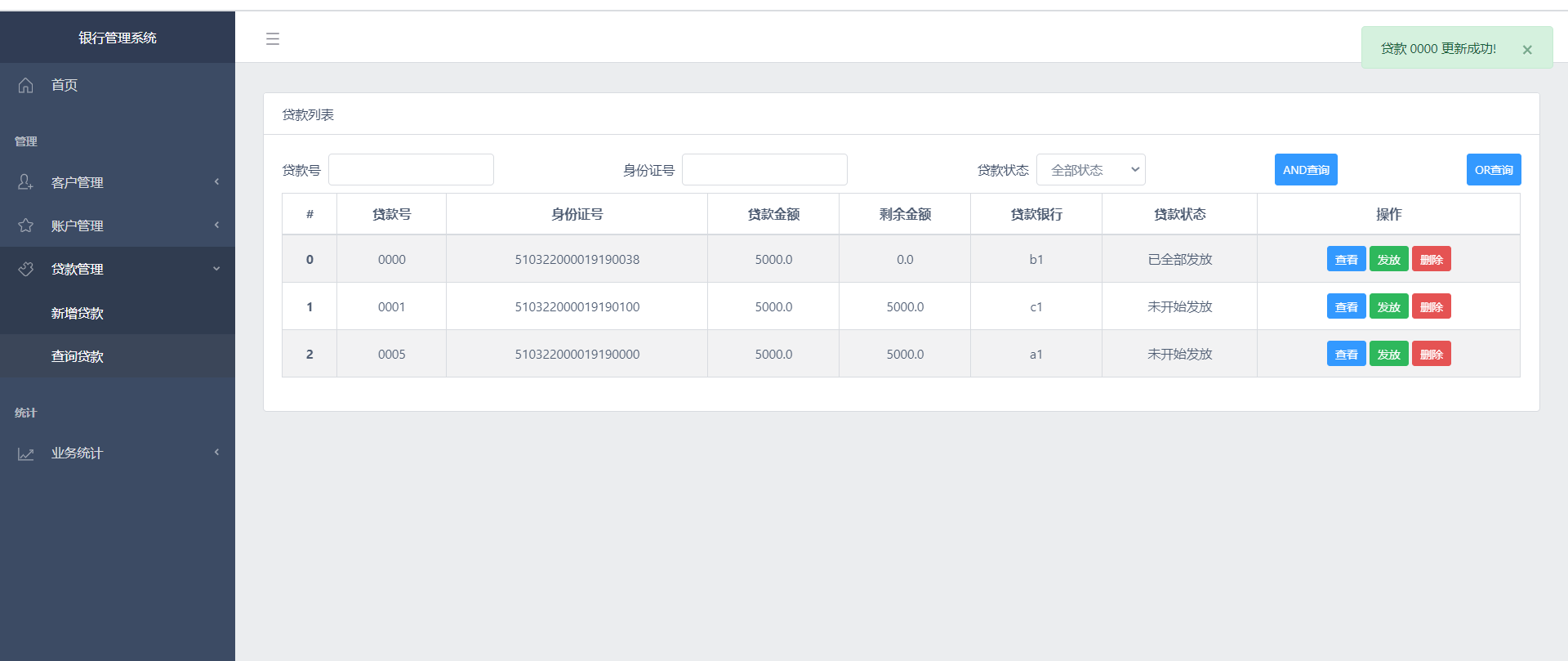
同理，AND和OR查询



查看对应贷款信息



点击发放按钮，会把未发放完的账户扣除相应的金额，比如把编号为0000的贷款发放500元，如下图



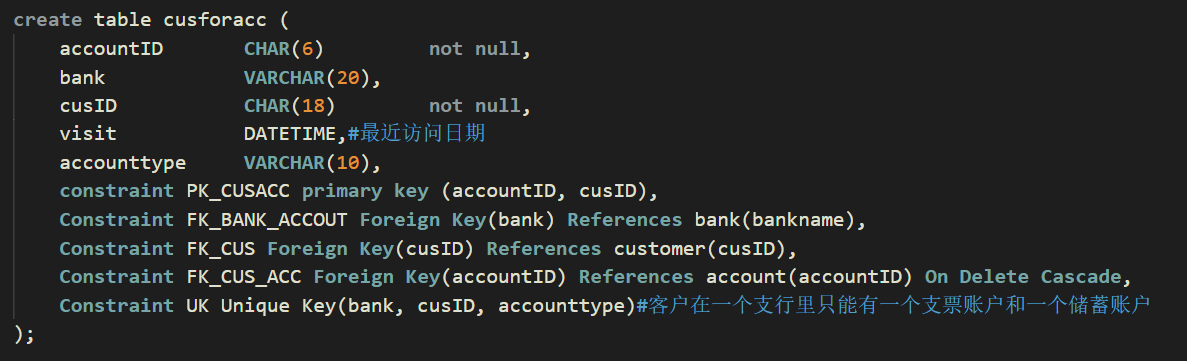
最后可以删除已全部发放完的贷款，比如刚才发放完的贷款0000



## 4.3 实现中的难点问题及解决【可选】

1. 首先是对账户的限制问题，一个客户在一个支行里只能有一个储蓄账户和一个支票账户

解决思路是建立cusforacc表，并在里面对bank、customer、accounttype使用一个unique key限制，这样一个支行里一个用户的账户数目就得到了限制



2. 具体写实验时其实代码框架有了之后是写起来是相对比较平稳的，主要难点还是在于对Flask、SQLAlchemy的学习，需要花费一定时间理解Web应用的开发流程，而且要和前端接口一一对应是相对繁琐的。

自己的想法就是面对比较新的一个领域，一定要先从顶层入手，对大致流程进行一个宏观上的学习，然后再细化到具体的应用或者项目，逐步完善，最后再测试、改一些bug。当然这个过程里调bug也是发现，自己预想的前端form提交的内容可能与实际的有差别，或者也可能用户使用系统时容易输错信息，所以有必要在后端接收时进行try except来避免系统崩溃。

3. 存在外键约束的表项删除或者插入一定要按顺序来

比如参照表里要插入一条数据时，那条数据的对应外键属性一定要在被参照表里有才行，所以应该先在被参照表里插入对应数据。删除时则相反，应该先把无外键约束（即参照表）删去，再删除被参照表的信息。

# 5 总结与讨论

经过本次实验，我对于数据库系统开发有了更深入的理解，完整经历了学习、设计、实现的过程，代码能力有了长足的提高，也学习到了Flask、SQLAlchemy等好用的框架，收获颇丰。

经验教训上，首先则是对新工具的学习方法，首先是要有一定的入门了解，知道实现系统的基本流程，每一步大概会用到一些什么，然后与实际应用、任务相结合，细化各个步骤，然后在具体实现的过程里去检索对应的文档，把功能跑出来。这样是一个比较好的开发方式，既减少了学习工具的遗忘成本，同时也不至于对工具一无所知，盲目做应用架构。

此外，在功能实现上，仍有一些细节被自己忽略，比如贷款是可以被多个客户所持有，这一点在检查实验时才发现自己漏考虑了。所以也是提醒自己需要对实验要求有更细致的阅读与把握。