

银行业务管理系统

系统设计与实现报告

姓名：张鹏飞

学号：PB17081521

计算机科学与技术学院

中国科学技术大学

2020 年 6 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.....	2
2	总体设计.....	3
2.1	系统模块结构.....	3
2.2	系统工作流程.....	4
2.3	数据库设计.....	4
3	详细设计.....	7
3.1	接口设计.....	7
3.1.1	外部接口.....	7
3.1.1.1	软件接口.....	7
3.1.1.2	硬件接口.....	7
3.1.1.3	用户接口.....	7
3.1.1.4	URL.....	7
3.1.2	内部接口.....	8
3.2	界面设计.....	10
3.2.1	主界面.....	11
3.2.2	支行界面.....	11
3.2.3	客户界面.....	13

3.2.4	贷款界面	14
3.2.5	账户界面	16
3.2.6	统计界面	18
3.3	数据库交互	19
3.3.1	数据库设计	19
3.3.2	ORM	19
3.3.3	数据库操作	100
4	实现与测试	222
4.1	实现结果	222
4.2	测试结果	255
4.2.1	客户管理	105
4.2.2	账户管理	107
4.2.3	贷款管理	30
4.2.4	业务统计	310
5	总结与讨论	33

1 概述

1.1 系统目标

开发一个基于 MySQL 数据库的银行业务管理系统

1.2 需求说明

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。

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，要求对统计结果同时提供表格和曲线图两种可视化展示方式。

1.3 本报告的主要贡献

向想要了解本应用开发的人系统地阐述设计者基于 MySQL 数据库的网页应用的开发流程，包括实现所用的语言及依赖、开发工具、开发思路、各模块的作用以及模块间的交互、端口开设与交互、数据库的设计、初始化、转换、通信等

2 总体设计

2.1 系统模块结构

```
bank\  
  |--assets\  
    |--//自己写的静态文件，会经过 webpack 打包，重新编译  
  |--database\  
    |--//与数据库交互、操作数据库的模块  
    |--__init__.py---//接口  
    |--model.py---//数据库模型  
    |--operate.py---//对数据库的各种操作  
  |--Routers\  
    |--//路由，控制端口间的通信  
    |--__init__.py---//接口  
  |--static\  
    |--//静态文件，编译时不受影响  
    |--bootstrap\  
      |--使用的 bootstrap 模板文件，可从官网找到并下载  
    |--css\  
      |--bootstrap.css  
      |--bootstrap.css.map  
      |--bootstrap.min.css  
      |--bootstrap.min.css.map  
      |--bootstrap-datetimepicker.min.css  
      |--bootstrap-theme.css  
      |--bootstrap-theme.css.map  
      |--bootstrap-theme.min.css  
      |--bootstrap-theme.min.css.map  
      |--ie10-viewport-bug-workaround.css  
      |--justified-nav.css  
    |--fonts\  
      |--glyphicons-halflings-regular.eot  
      |--glyphicons-halflings-regular.svg  
      |--glyphicons-halflings-regular.ttf  
      |--glyphicons-halflings-regular.woff  
      |--glyphicons-halflings-regular.woff2  
    |--js\  
      |--bootstrap.js  
      |--bootstrap.min.js  
      |--bootstrap-datetimepicker.min.js  
      |--jquery.min.js
```

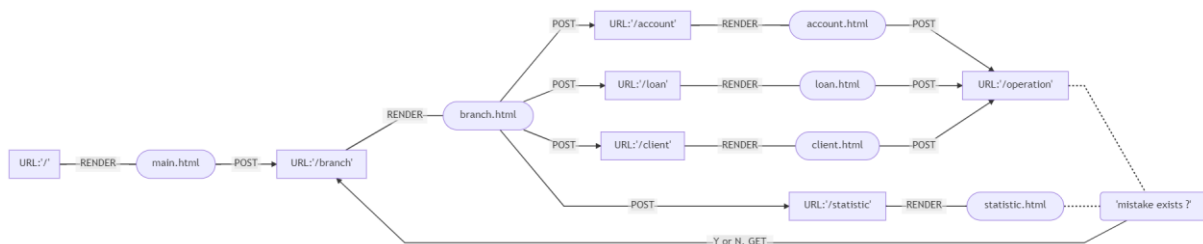
```

|---npm.js
|---favicon.ico---//网页图标
|---lab3.sql---//仅用于展示 sql 设计，应用运行中不起作用
|---templates
|   |---account.html---//账户操作界面
|   |---branch.html---//支行操作界面
|   |---client.html---//客户操作界面
|   |---loan.html---//贷款操作界面
|   |---main.html---//主界面
|   |---statistic.html---//统计界面
|---app.py---//运行文件
|---README.md---//使用方法
|---requirements.txt---//依赖

```

2.2 系统工作流程

各模块间通信方式如下



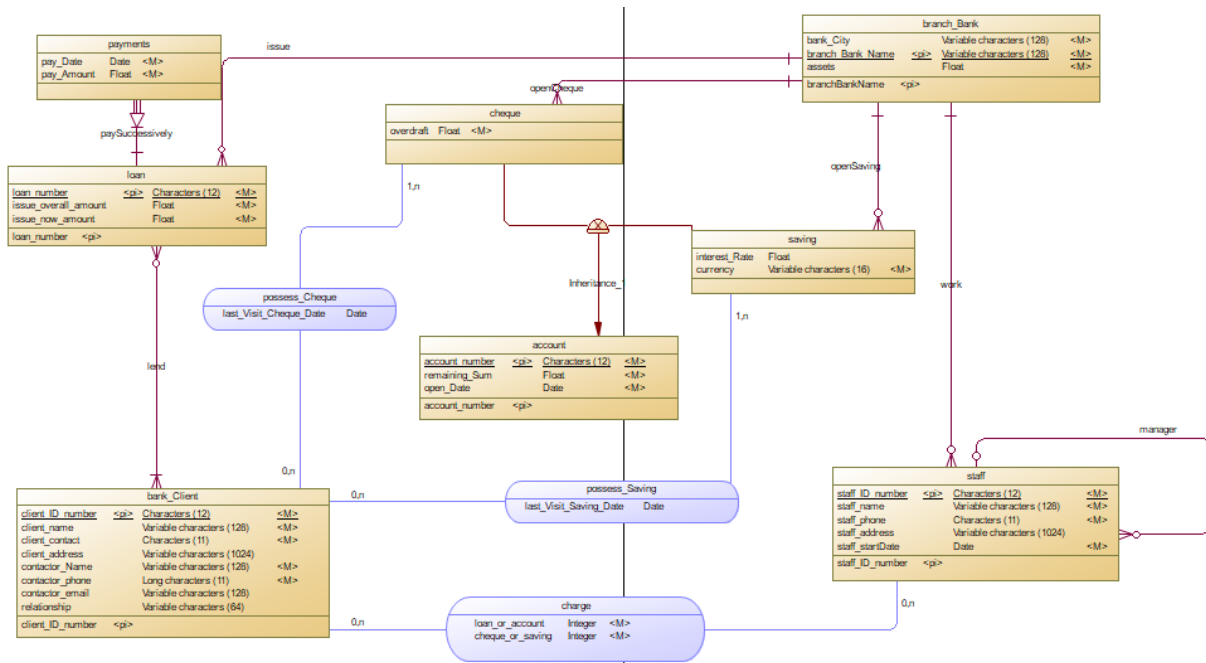
除图示方式之外，各模板与各接口间还可通过给定的交互式按钮进行 GET 请求的方式通信

2.3 数据库设计

使用 powerdesign16 工具设计

从 ER 图入手，生成物理数据库，对其修改、优化后生成 mysql 的 .sql 文件，再向其中插入所需的初始数据，在本应用中，为银行信息、员工信息；员工通过员工号区分部门，某员工号与经理号相同时该员工为经理

ER 图设计



物理数据库设计

在 ER 图生成的原始物理数据库中做如下修改：

1. 为满足"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"

1.1 charge 表添加外键(FK_CHARGE_REFERENCE_BANK_NAM)引用 branch bank

中的 branchBankName，相应的，表中多出了 branchBanckName 属性

1.2 charge 表添加外键(FK_CHARGE_REFERENCE_BANK_NAM)引用 branch bank

中的 branchBankName，相应的，表中多出了 branchBanckName 属性

1.3 branchBankName、chargeType 设为 mandatory，chargeType 更名为

loan_or_account，添加 cheque_or_saving 属性

1.4 添加 unique key，属性包括 branchBankName，cheque_or_saving，

client_ID_number

1.5 CHARGE_TYPE 设为 CHARGE 表的主键

2. 为满足"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"

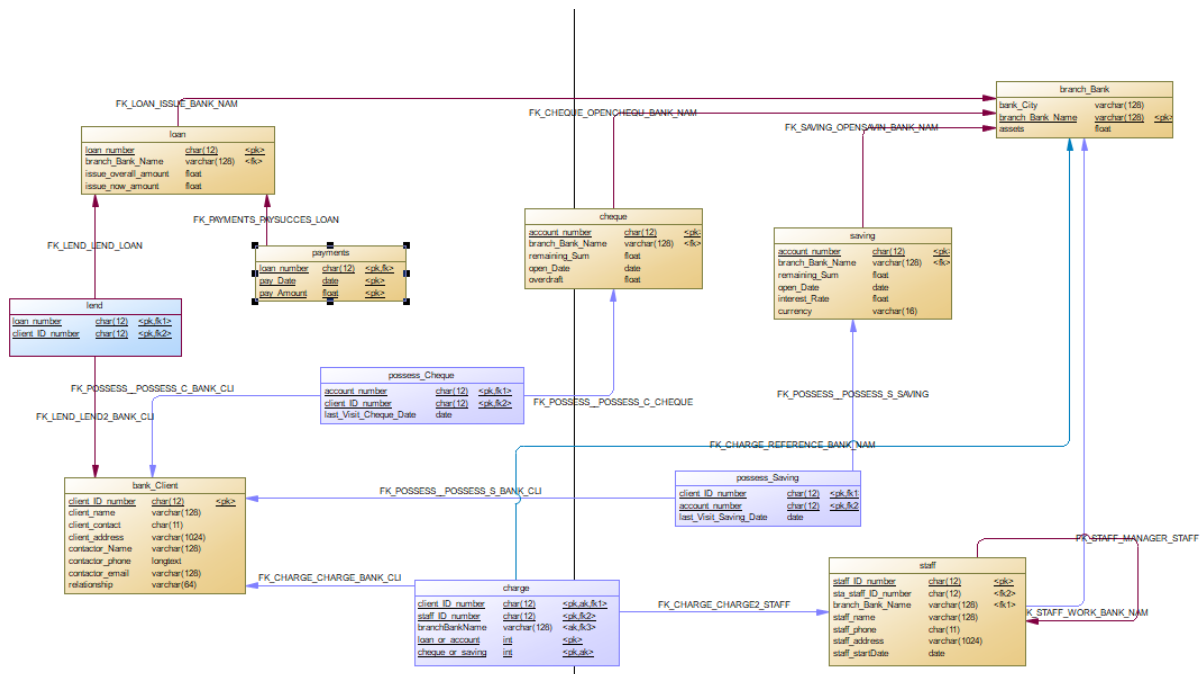
2.1 将 payments 的 payDate 属性设置为另一个主键

2.2 为简化设计并防止与 Charge 表冲突，限定一个客户在一个支行中仅可以贷款一次

3. 为每个表加上 charset = UTF8 的属性

4. 父表不为 branch_Bank, staff 的外键引用，需要加上 on delete cascade

最终的物理数据库设计为：



初始数据插入

在生成的 sql 文件中，执行向 branch_Bank, staff 插入已准备数据的语句

3 详细设计

3.1 接口设计

3.1.1 外部接口

3.1.1.1 软件接口

本项目的后端必须部署在支持 MySQL 的应用数据库服务器上，前端应能支持 HTML 网页展示

3.1.1.2 硬件接口

本项目的后端使用的数据库管理系统为 MySQL，服务器端基于 HTML 网页。

3.1.1.3 用户接口

本系统对用户提供的网页图形接口，用户使用浏览器可对图形化界面进行操作。

3.1.1.4 URL

URL	描述	请求方式	用户权限
http://127.0.0.1:5000/	主界面所在的端口	GET	可访问
http://127.0.0.1:5000/branch	支行管理界面所在的端口	GET, POST	可访问
http://127.0.0.1:5000/client	客户管理界面所在的端口	GET, POST	可访问
http://127.0.0.1:5000/loan	贷款管理界面所在的端口	GET, POST	可访问
http://127.0.0.1:5000/account	账户管理界面所在的端口	GET, POST	可访问
http://127.0.0.1:5000/statistic	统计界面所在的端口	GET, POST	可访问
http://127.0.0.1:5000/operation	数据库交互所在的端口	POST	不可访问

3.1.2 内部接口

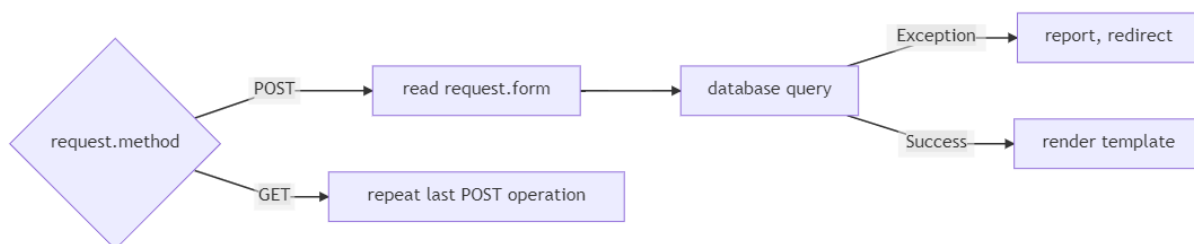
1. app.py

➤ main_router()

—输入：<http://127.0.0.1:5000/>上的请求

—输出：渲染 main.html

—流程图：



➤ branch()

—输入：<http://127.0.0.1:5000/branch>上的请求

—输出：渲染 branch.html

—流程图：同 `main_router()`

➤ client()

—输入：<http://127.0.0.1:5000/client>上的请求

—输出：渲染 client.html

—流程图：同 `main_router()`

➤ account()

—输入：<http://127.0.0.1:5000/account>上的请求

—输出：渲染 account.html

—流程图：同同 `main_router()`

➤ loan()

—输入：<http://127.0.0.1:5000/loan>上的请求

—输出：渲染 loan.html

—流程图：同 main_router()

➤ statistic()

—输入：<http://127.0.0.1:5000/statistic>上的请求

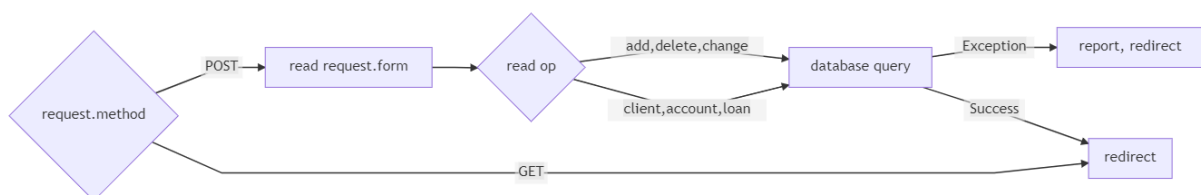
—输出：statistic.html

➤ operation()

—输入：<http://127.0.0.1:5000/operation>上的请求

—输出：重定向到 <http://127.0.0.1:5000/branch>

—流程图：

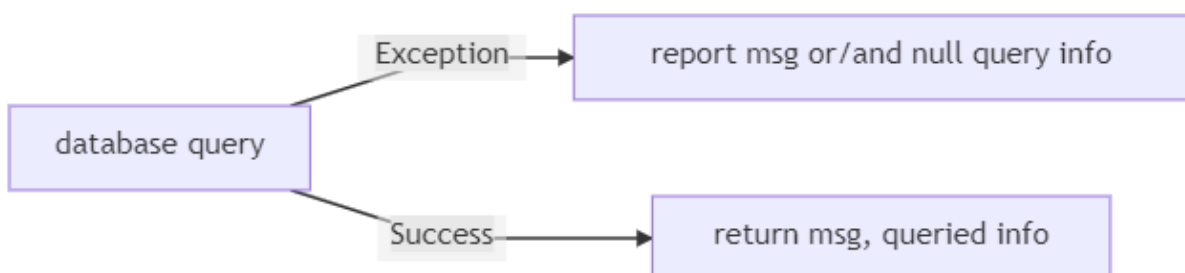


2. database.operate.py

➤ LoanOperate ()

—贷款操作类，初始化：branch_name 支行名，handle_loan 贷款号

—各方法流程图概括：



➤ ClientOperate ()

—客户操作类，初始化：branch_name 支行名，handle_client 身份证号

—各方法流程图概括：同 LoanOperate ()

➤ AccountOperate ()

—账户操作类，初始化：branch_name 支行名，handle_account 账户号

—各方法流程图概括：同 LoanOperate ()

➤ Overall ()

—全局查询类，初始化：branch_name 支行名

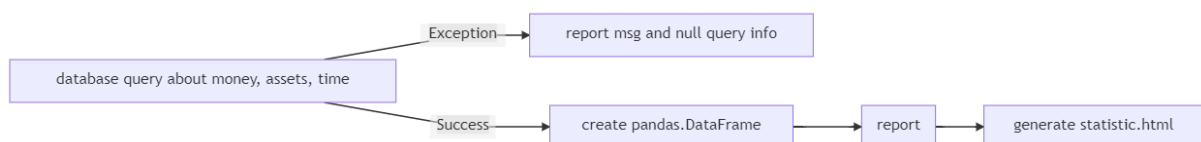
—各方法流程图概括：同 LoanOperate ()

➤ generate_dataframe ()

—输入：start_time 开始时间, branch_name 支行名, sta_select 时间跨度选择

—输出：statistic.html

—流程图：



3.2 界面设计

前端采用 Bootstrap 模板，使用 HTML，CSS，JavaScript 进行初始布局

通过表单向后端发送数据

对于发送到模板的数据，使用 Jinja2 模板进行渲染

3.2.1 主界面

显示所有支行的所有信息，用户通过其后的单选按钮选择进入某支行

银行管理界面

支行管理

支行名	所在城市	资产	查询/管理
上海支行	上海	10053000.0	<input type="radio"/>
北京支行	北京	10000000.0	<input type="radio"/>
合肥一支行	合肥	10000000.0	<input type="radio"/>
合肥二支行	合肥	10000000.0	<input type="radio"/>

© Pengfei Zhang, All Rights Reserved.

3.2.2 支行界面

每个支行的管理主界面，其中包含客户、账户、贷款的较少的识别性基础信息，同时也是查询该支行统计信息的唯一面板

用户通过单选按钮选择查看具体信息，或通过添加按钮选择增加一个客户、账户、贷款等；为确保原子性，交互式接口（单选按钮、表单提交按钮）强制用户一次仅能对一项内容的一个条目进行操作；某些固定输入（如日期输入）则通过 jquery 限制了用户操作与输入格式

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

业务客户信息

客户姓名	身份证号	联系人手机	查询修改
李明不	ID343643	86087654321	<input type="radio"/>
郑超	ID465463	86108765432	<input type="radio"/>
李明明	ID534656	86876543210	<input type="radio"/>
添加新客户			<input type="radio"/>

账户信息

开户支行	账户号	账户类型	查询修改
上海支行	A353553	支票	<input type="radio"/>
上海支行	A355676	支票	<input type="radio"/>
上海支行	A234234	储蓄	<input type="radio"/>
上海支行	A245656	储蓄	<input type="radio"/>
添加新账户			<input type="radio"/>

贷款信息

贷款状态	贷款号	发放支行	查询修改
已全部发放	L456678	上海支行	<input type="radio"/>
发放中	L567567	上海支行	<input type="radio"/>
添加新贷款			<input type="radio"/>

业务统计

提交

按日查询

按季查询

按年查询

开始日期

2020-06-03

查询

← June 2020 →

Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

© Pengfei Zhang, All Rights Reserved

最顶部的导航栏提供了查看其他界面的通道，可查看上次操作时的客户、账户、贷款、统计信息。值得注意的是，我们不建议在刚进入支行未进行对应操作时这么做，因为此时不存在“上次操作”。“查询其他支行”按钮提供了通往主界面的通道

红色框为警告栏。当有错误发生时，会强制跳转到该页面，并在警告栏中打印出错信息。如果是输入信息有错，用户可根据信息提示更正自己的输入；如果是系统错误，用户可及时联系管理员

3.2.3 客户界面

对客户进行管理的界面。当支行界面的信息为查询某客户时，渲染客户的所有信息、拥有账户及贷款、信息修改部分；否则支行界面的信息为添加一个客户，渲染添加客户的部分

添加客户界面

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

客户信息

提交

增加

客户姓名	身份证号	联系电话	地址
啊啊啊	ID534656	86012345678	南京市A街A路A号

联系人姓名	与客户关系	联系人电话	联系人邮箱
啊啊啊	父亲	86876543210	876543210@mail.cn

业务类型

负责员工

储蓄账户

支票账户

贷款

客户信息查询/修改界面

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

客户信息

提交

删除

修改

客户姓名	身份证号	联系电话	地址
哪啊不	ID343643	86123456780	南京市A街A路B号

客户姓名

联系电话

地址

联系人姓名	与客户关系	联系人电话	联系人邮箱
不啊哪	父亲	86087654321	087654321@mail.cn

联系人姓名

父亲

联系人电话

联系人邮箱

业务类型	负责员工
储蓄账户	ID111101
支票账户	ID111101
贷款	ID111101

3.2.4 贷款界面

对贷款进行管理的界面。当支行界面的信息为查询某贷款时，渲染贷款的所有信息、拥有客户、每次款项的部分；否则支行界面的信息为添加一个贷款，渲染添加贷款的部

分。添加贷款时，我们要求一次性添加所有的付款信息，到达某时刻自动付款

添加贷款界面

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

贷款信息

提交

增加

贷款号	发放总次数	发放总金额
L131454	2	5000

账户客户信息

客户身份证号	权限
ID343643	<input checked="" type="checkbox"/>
ID465463	<input type="checkbox"/>
ID534656	<input type="checkbox"/>

贷款信息

贷款发放日期	发放金额(¥)	
2020-07-01	2000	<div>+ 删除</div>
2020-09-08	3000	<div>+ 删除</div>

点击加号可自动增加条目框，多余的框请删除；下方需勾选持有该贷款的客户

贷款信息查询/删除界面

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

贷款信息

提交

删除

贷款号	发放总次数	发放总金额
L567567	2	30000.0

账户客户信息

客户身份证号	权限
ID343643	持有

贷款信息

贷款发放日期	发放金额(¥)	已发放
2020-06-09	10000.0	YES
2020-07-09	20000.0	NO

3.2.5 账户界面

对账户进行管理的界面。当支行界面的信息为查询某客户时，渲染账户的所有信息、拥有客户、客户访问时间；否则支行界面的信息为添加一个账户，渲染添加账户的部分。

开户时，需要选择账户类型，多余的值会被我们抛弃，其余值不可缺少；下方有所有客户的信息，用于勾选拥有此账户的客户

修改时，输入需要修改的信息提交即可，非法值会被我们抛弃；我们目前支持多个信息的联合修改，以及多个客户最近访问时间的联合修改。

添加账户界面

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

账户信息

提交

开户

账户号	开户日期	账户类型
<div>A067607</div>	<div>2020-04-02</div>	<div>支票账户</div>

余额	透支额	利率	货币类型
<div>1835</div>	<div>20000</div>	<div>-1</div>	<div>人民币</div>

账户客户信息

客户身份证号	最近访问日期	权限	修改
ID343643		不持有	<input checked="" type="checkbox"/>
ID465463		不持有	<input type="checkbox"/>
ID534656		不持有	<input checked="" type="checkbox"/>

账户查询/修改界面

支行管理

客户管理

账户管理

贷款管理

业务统计

您正在查询的支行：上海支行

查询其他支行

账户信息

提交

销户

修改

账户号	开户日期	账户类型
A067607	支票账户	2020-04-02

余额	透支额	利率	货币类型
1835.0	20000.0		

余额

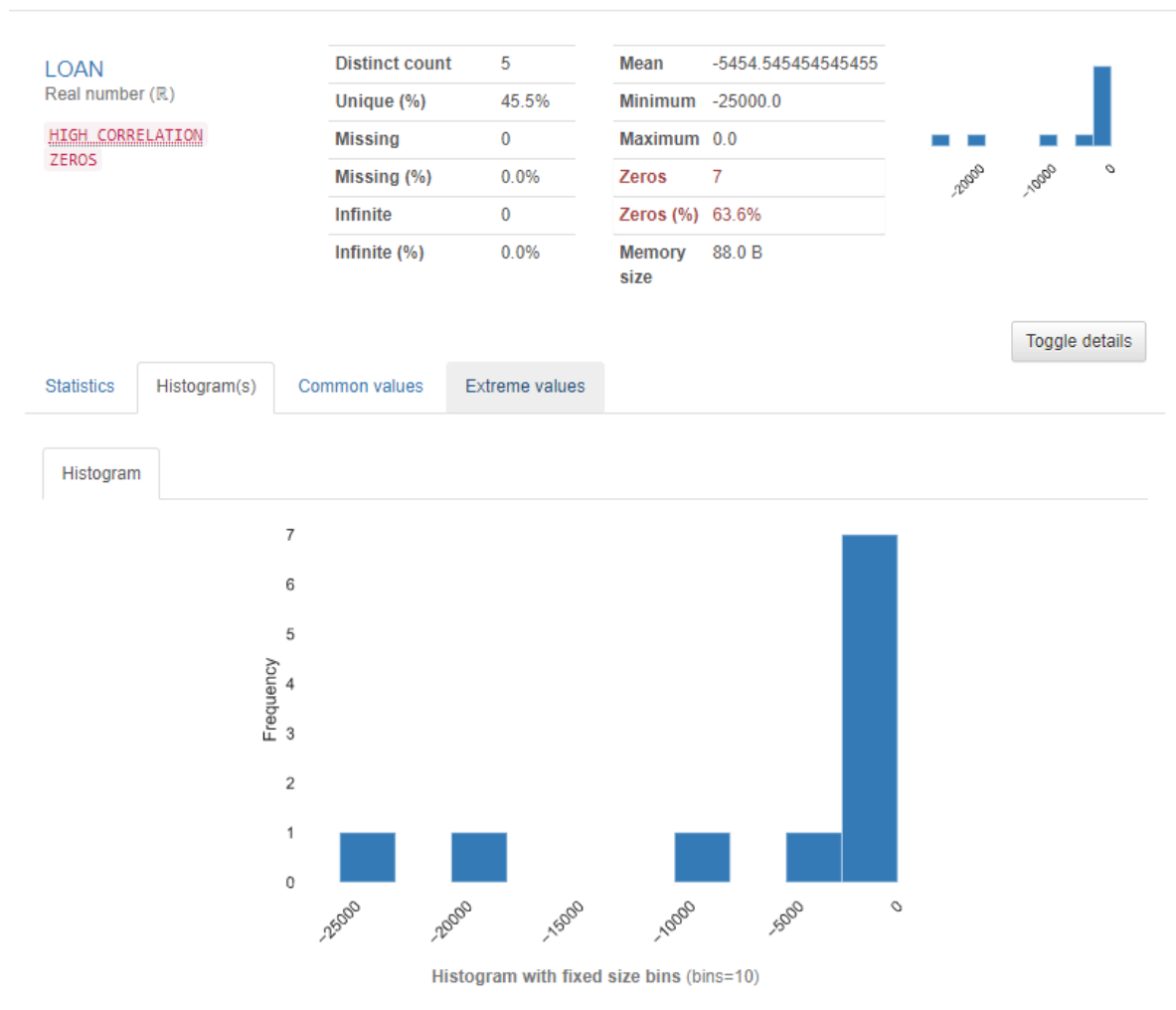
透支额

账户客户信息

客户身份证号	最近访问日期	权限	修改
ID343643	2020-04-02 访问日期	持有	<input checked="" type="checkbox"/>
ID534656	2020-04-02 访问日期	持有	<input checked="" type="checkbox"/>
ID465463		不持有	<input type="checkbox"/>

3.2.6 统计界面

数据库读取各时间段内的申请账户数、申请客户数、贷款发放数、银行资产变化，数据以 pandas.DataFrame 的形式存储采用 pandas-profiling 自动生成带图表的 HTML 文件



3.2 数据库交互

3.2.1 数据库设计

具体设计见 lab2，对 lab2 的补充见 2.3

3.2.2 ORM

使用 ORM 模型的 peewee 库将数据库中的表、行、列分别映射成 python 中的类、属性、记录(对象)，便于操作数据库

对象关系映射（Object Relational Mapping，简称 ORM）模式是为了解决面向对象与关系数据库存在的互不匹配的现象的技术。简单的说，ORM 是通过使用描述对象和数据库之间映射的元数据，将程序中的对象自动持久化到关系数据库中。那么，到底如何实现持久化呢？一种简单的方案是采用硬编码方式，为每一种可能的数据库访问操作提供单独的方法。

这种方案存在以下不足：

1. 持久化层缺乏弹性。一旦出现业务需求的变更，就必须修改持久化层的接口
2. 持久化层同时与域模型与关系数据库模型绑定，不管域模型还是关系数据库模型发生变化，毒药修改持久化层的相关程序代码，增加了软件的维护难度。

ORM 提供了实现持久化层的另一种模式，它采用映射元数据来描述对象关系的映射，使得 ORM 中间件能在任何一个应用的业务逻辑层和数据库层之间充当桥梁。

ORM 的方法论基于三个核心原则：

- 简单：以最基本形式建模数据。
- 传达性：数据库结构被任何人都能理解的语言文档化。
- 精确性：基于数据模型创建正确标准化了的结构

具体方法见 README.md

3.2.3 数据库操作

增删查改

值得提起的是，ORM 是基于固定数据库的，因此在使用时需要确保数据库连接正常，且所交互的数据库存在，需要提前生成

增加：Model.create()

删除：Model.delete().execute()

查询：Model.select().get()返回单条查询记录，或 Model.select().execute()返回查询结果的游标

修改：先查询后修改，然后保存

这些函数的语法均是基于 MySQL 的 SELECT, INSERT, DELETE 语句；如果希望在查询过程中使用 MySQL 的函数，需使用 fn 对象

具体操作

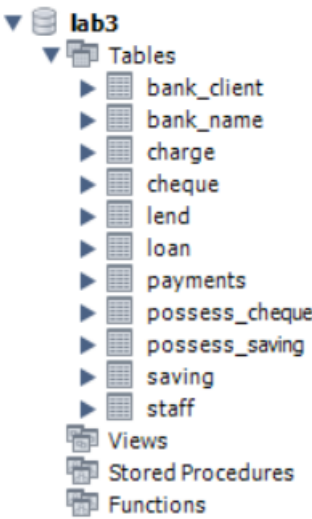
处理对象	增加	删除	修改
储蓄账户	按顺序插入 saving 表, possess_saving 表, charge 表, 修改 bank_name 表	删除 saving 中的条目, 视情况删除 charge 中的条目, 修改 bank_name 表	修改 saving 表或 possess_saving 表, 视情况修改 bank_name 表
支票账户	按顺序插入 cheque 表, possess_cheque 表, charge 表, 修改 bank_name 表	删除 cheque 中的条目, 视情况删除 charge 中的条目, 修改 bank_name 表	修改 cheque 表或 possess_cheque 表, 视情况修改 bank_name 表
客户	插入 bank_client 表	删除 bank_client 中的条目	修改 bank_client 表
贷款	按顺序插入 loan 表, lend 表, payments 表, charge 表, 修改 bank_name 表	删除 loan 中的条目, 视情况删除 charge 中的条目, 修改 bank_name 表	

4 实现与测试

4.1 实现结果

前端结果已在前面给出，这里用 MySQL Workbench 展示数据库中的结果

数据库：



bank_name 表：

	BANK_CITY	BRANCH_BANK_NAME	ASSETS
▶	上海	上海支行	10072800
	北京	北京支行	10000000
	合肥	合肥一支行	10000000
	合肥	合肥二支行	10000000
*	NULL	NULL	NULL

bank_client 表：

	CLIENT_ID_NUMBER	CLIENT_NAME	CLIENT_CONTACT	CLIENT_ADDRESS	CONTACTOR_NAME	CONTACTOR_PHONE	CONTACTOR_EMAIL	RELATIONSHIP
▶	ID343643	嘟嘟不	86123456780	南京市A街A路B号	不嘟嘟	86087654321	087654321@mail.cn	父亲
	ID465463	那把	86234567801	南京市A街B路A号	把那	86108765432	108765432@mail.cn	母亲
	ID534656	嘟嘟啊	86012345678	南京市A街A路A号	啊嘟嘟	86876543210	876543210@mail.cn	父亲
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

charge 表：

	CLIENT_ID_NUMBER	STAFF_ID_NUMBER	BRANCHBANKNAME	LOAN_OR_ACCOUNT	CHEQUE_OR_SAVING
▶	ID343643	ID111101	上海支行	0	0
	ID343643	ID111101	上海支行	2	1
	ID343643	ID111101	上海支行	2	2
	ID465463	ID110001	上海支行	0	0
	ID465463	ID110001	上海支行	1	3
	ID465463	ID110001	上海支行	2	1
	ID465463	ID110001	上海支行	2	2
	ID534656	ID110001	上海支行	0	0
	ID534656	ID110001	上海支行	2	1
	ID534656	ID110001	上海支行	2	2
*	NULL	NULL	NULL	NULL	NULL

cheque 表：

	ACCOUNT_NUMBER	BRANCH_BANK_NAME	REMAINING_SUM	OPEN_DATE	OVERDRAFT
▶	A067607	上海支行	1835	2020-04-02	20000
	A355676	上海支行	1000	2020-01-11	10000
*	NULL	NULL	NULL	NULL	NULL

lend 表：

	LOAN_NUMBER	CLIENT_ID_NUMBER
▶	L567567	ID343643
	L909089	ID465463
*	NULL	NULL

loan 表：

	LOAN_NUMBER	BRANCH_BANK_NAME	ISSUE_OVERALL_AMOUNT	ISSUE_NOW_AMOUNT
▶	L567567	上海支行	30000	10000
	L909089	上海支行	30000	30000
*	NULL	NULL	NULL	NULL

payments 表：

	LOAN_NUMBER	PAY_DATE	PAY_AMOUNT
▶	L567567	2020-06-09	10000
	L567567	2020-07-09	20000
	L909089	2020-02-14	25000
	L909089	2020-03-14	5000
*	NULL	NULL	NULL

possess_cheque 表：

	ACCOUNT_NUMBER	CLIENT_ID_NUMBER	LAST_VISIT_CHEQUE_DATE
▶	A067607	ID343643	2020-04-02
	A067607	ID534656	2020-04-02
	A355676	ID465463	2020-01-11
*	NULL	NULL	NULL

possess_saving 表：

	CLIENT_ID_NUMBER	ACCOUNT_NUMBER	LAST_VISIT_SAVING_DATE
▶	ID343643	A234234	2019-12-27
	ID465463	A245656	2019-08-08
	ID534656	A234234	2019-12-27
★	NULL	NULL	NULL

saving 表：

	ACCOUNT_NUMBER	BRANCH_BANK_NAME	REMAINING_SUM	OPEN_DATE	INTEREST_RATE	CURRENCY
▶	A234234	上海支行	10000	2019-12-27	0.03	rmb
	A245656	上海支行	100000	2019-08-08	0.12	rmb
★	NULL	NULL	NULL	NULL	NULL	NULL

staff 表：

	STAFF_ID_NUMBER	STA_STAFF_ID_NUMBER	BRANCH_BANK_NAME	STAFF_NAME	STAFF_PHONE	STAFF_ADDRESS	STAFF_STARTDATE
▶	ID000000	ID000000	北京支行	吧啊啊	86000000000	北京市A街A路A号	2008-08-09
	ID000001	ID000000	北京支行	把阿炳	86000000001	北京市A街A路B号	2008-08-09
	ID000002	ID000000	北京支行	吧啊擦	86000000002	北京市A街A路C号	2008-08-10
	ID001100	ID001100	北京支行	巴巴	86001100000	北京市A街B路A号	2008-08-19
	ID001101	ID001100	北京支行	八部半	86001100001	北京市A街B路B号	2008-12-09
	ID110000	ID110000	上海支行	撒啊啊	86110000000	上海市A街A路A号	2008-11-09
	ID110001	ID110000	上海支行	Saab	86110000001	上海市A街A路B号	2009-02-09
	ID111100	ID111100	上海支行	撒把	86111100000	上海市A街B路A号	2009-08-09
	ID111101	ID111100	上海支行	萨博吧	86111100001	上海市A街B路B号	2009-10-10
	ID220000	ID220000	合肥一支行	哈啊啊	86220000000	合肥市A街A路A号	2009-12-09
	ID220001	ID220000	合肥一支行	哈啊不	86220000001	合肥市A街A路B号	2009-01-09
	ID221100	ID221100	合肥一支行	哈巴	86221100000	合肥市A街B路A号	2009-02-10
	ID221101	ID221100	合肥一支行	哈比比	86221100001	合肥市A街B路B号	2010-08-29
	ID222200	ID222200	合肥一支行	哈擦	86222200000	合肥市A街C路A号	2010-06-05
	ID222201	ID222200	合肥一支行	哈擦不	86222200001	合肥市A街C路B号	2011-01-04
	ID330000	ID330000	合肥二支行	好吧啊	86330000000	合肥市B街A路A号	2011-07-02
	ID330001	ID330000	合肥二支行	海拔表	86330000001	合肥市B街A路B号	2012-08-25
	ID330002	ID330000	合肥二支行	后拔出	86330000002	合肥市B街A路C号	2012-09-03
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

可对照网页的显示信息核查

4.2 测试结果

4.2.1 客户管理

增加

客户信息

提交

增加

客户姓名

身份证号

联系电话

地址

那成本

ID235346

86345678012

南京市A街C路B号

联系人姓名

与客户关系

联系人电话

联系人邮箱

本成那

父亲

86210876543

210876543@mail.cn

查看

客户信息

提交

删除

修改

客户姓名

身份证号

联系电话

地址

那成本

ID235346

86345678012

南京市A街C路B号

客户姓名

联系电话

地址

联系人姓名

与客户关系

联系人电话

联系人邮箱

本成那

父亲

86210876543

210876543@mail.cn

联系人姓名

父亲

联系人电话

联系人邮箱

业务类型

负责员工

储蓄账户

支票账户

贷款

修改

客户信息

提交

删除

修改

客户姓名

身份证号

联系电话

地址

那成本

ID235346

86345678012

南京市A街C路B号

客户姓名

联系电话

地址

联系人姓名

与客户关系

联系人电话

联系人邮箱

本成那

母亲

86210876543

210876543@mail.cn.com

联系人姓名

父亲

联系人电话

联系人邮箱

删除

业务客户信息

客户姓名	身份证号	联系人手机	查询/修改
那成本	ID235346	86210876543	<input type="radio"/>
哪啊不	ID343643	86087654321	<input type="radio"/>
那把	ID465463	86108765432	<input type="radio"/>
哪啊啊	ID534656	86876543210	<input type="radio"/>
添加新客户			<input type="radio"/>

客户姓名	身份证号	联系人手机	查询/修改
哪啊不	ID343643	86087654321	<input type="radio"/>
那把	ID465463	86108765432	<input type="radio"/>
哪啊啊	ID534656	86876543210	<input type="radio"/>
添加新客户			<input type="radio"/>

异常处理

支行管理	客户管理	账户管理	贷款管理	业务统计
------	------	------	------	------

您正在查询的支行：上海支行

查询其他支行

出现错误IntegrityError(1062, "Duplicate entry 'ID343643' for key 'bank_client.PRIMARY'")

业务客户信息

客户姓名	身份证号	联系人手机	查询/修改
哪啊不	ID343643	86087654321	<input type="radio"/>
那把	ID465463	86108765432	<input type="radio"/>
哪啊啊	ID534656	86876543210	<input type="radio"/>
添加新客户			<input type="radio"/>

如果是输入信息不合法，红框仅会显示“出现错误”

4.2.2 账户管理

插入

账户信息

提交 开户

账户号	开户日期	账户类型
<input type="text" value="A897890"/>	<input type="text" value="2020-01-17"/>	<input type="text" value="支票账户"/>

余额	透支额	利率	货币类型
<input type="text" value="2000"/>	<input type="text" value="10000"/>	<input type="text" value="-1"/>	<input type="text" value="人民币"/>

账户客户信息

客户身份证号	最近访问日期	权限	修改
ID343643		不持有	<input type="checkbox"/>
ID465463		不持有	<input checked="" type="checkbox"/>
ID534656		不持有	<input type="checkbox"/>

查询

账户信息

提交 销户 修改

账户号	开户日期	账户类型
A897890	支票账户	2020-01-17

余额	透支额	利率	货币类型
2000.0	10000.0		

账户客户信息

客户身份证号	最近访问日期	权限	修改
ID465463	2020-01-17 <input type="text" value="访问日期"/>	持有	<input checked="" type="checkbox"/>
ID343643		不持有	<input type="checkbox"/>
ID534656		不持有	<input type="checkbox"/>

修改

账户信息

提交

销户

修改

账户号	开户日期	账户类型
A897890	支票账户	2020-01-17

余额	透支额	利率	货币类型
1980.0	10000.0		

余额

透支额

账户客户信息

客户身份证号	最近访问日期	权限	修改
ID465463	2020-02-15 <div>访问日期</div>	持有	<input checked="" type="checkbox"/>
ID343643		不持有	<input type="checkbox"/>
ID534656		不持有	<input type="checkbox"/>

删除

账户信息

开户支行	账户号	账户类型	查询/修改
上海支行	A067607	支票	<input type="radio"/>
上海支行	A897890	支票	<input type="radio"/>
上海支行	A234234	储蓄	<input type="radio"/>
上海支行	A245656	储蓄	<input type="radio"/>
添加新账户			<input type="radio"/>

账户信息

开户支行	账户号	账户类型	查询/修改
上海支行	A067607	支票	<input type="radio"/>
上海支行	A234234	储蓄	<input type="radio"/>
上海支行	A245656	储蓄	<input type="radio"/>
添加新账户			<input type="radio"/>

异常处理

您正在查询的支行：上海支行

查询其他支行

出现错误唯一键检查出错

业务客户信息

客户姓名	身份证号	联系人手机	查询/修改
哪啊不	ID343643	86087654321	○
那把	ID465463	86108765432	○
哪啊啊	ID534656	86876543210	○
添加新客户			○

账户信息

开户支行	账户号	账户类型	查询/修改
上海支行	A067607	支票	○
上海支行	A355676	支票	○
上海支行	A234234	储蓄	○
上海支行	A245656	储蓄	○
添加新账户			○

4.2.3 贷款管理

增加

贷款信息

提交 增加

贷款号	发放总次数	发放总金额
L456456	4	100000

账户客户信息

客户身份证号	权限
ID343643	<input type="checkbox"/>
ID465463	<input type="checkbox"/>
ID534656	<input checked="" type="checkbox"/>

贷款信息

贷款发放日期	发放金额(¥)	
2020-01-16	40000	<div>+</div> <div>删除</div>
2020-02-16	30000	<div>删除</div>
2020-03-16	20000	<div>删除</div>
2020-08-20	10000	<div>删除</div>

查询

贷款信息

提交 删除

贷款号	发放总次数	发放总金额
L456456	4	100000.0

账户客户信息

客户身份证号	权限
ID534656	持有

贷款信息

贷款发放日期	发放金额(¥)	已发放
2020-01-16	40000.0	YES
2020-02-16	30000.0	YES
2020-03-16	20000.0	YES
2020-08-20	10000.0	NO

删除

贷款信息

贷款状态	贷款号	发放支行	查询/修改
发放中	L456456	上海支行	<input type="radio"/>
发放中	L567567	上海支行	<input type="radio"/>
添加新贷款			<input type="radio"/>

异常处理

您正在查询的支行：上海支行

查询其他支行

出现错误不允许删除

业务客户信息

客户姓名	身份证号	联系人手机	查询/修改
哪啊不	ID343643	86087654321	<input type="radio"/>
那把	ID465463	86108765432	<input type="radio"/>
哪啊啊	ID534656	86876543210	<input type="radio"/>
添加新客户			<input type="radio"/>

账户信息

开户支行	账户号	账户类型	查询/修改
上海支行	A067607	支票	<input type="radio"/>
上海支行	A897890	支票	<input type="radio"/>
上海支行	A234234	储蓄	<input type="radio"/>
上海支行	A245656	储蓄	<input type="radio"/>
添加新账户			<input type="radio"/>

贷款信息

贷款状态	贷款号	发放支行	查询/修改
发放中	L456456	上海支行	<input type="radio"/>
发放中	L567567	上海支行	<input type="radio"/>
添加新贷款			<input type="radio"/>

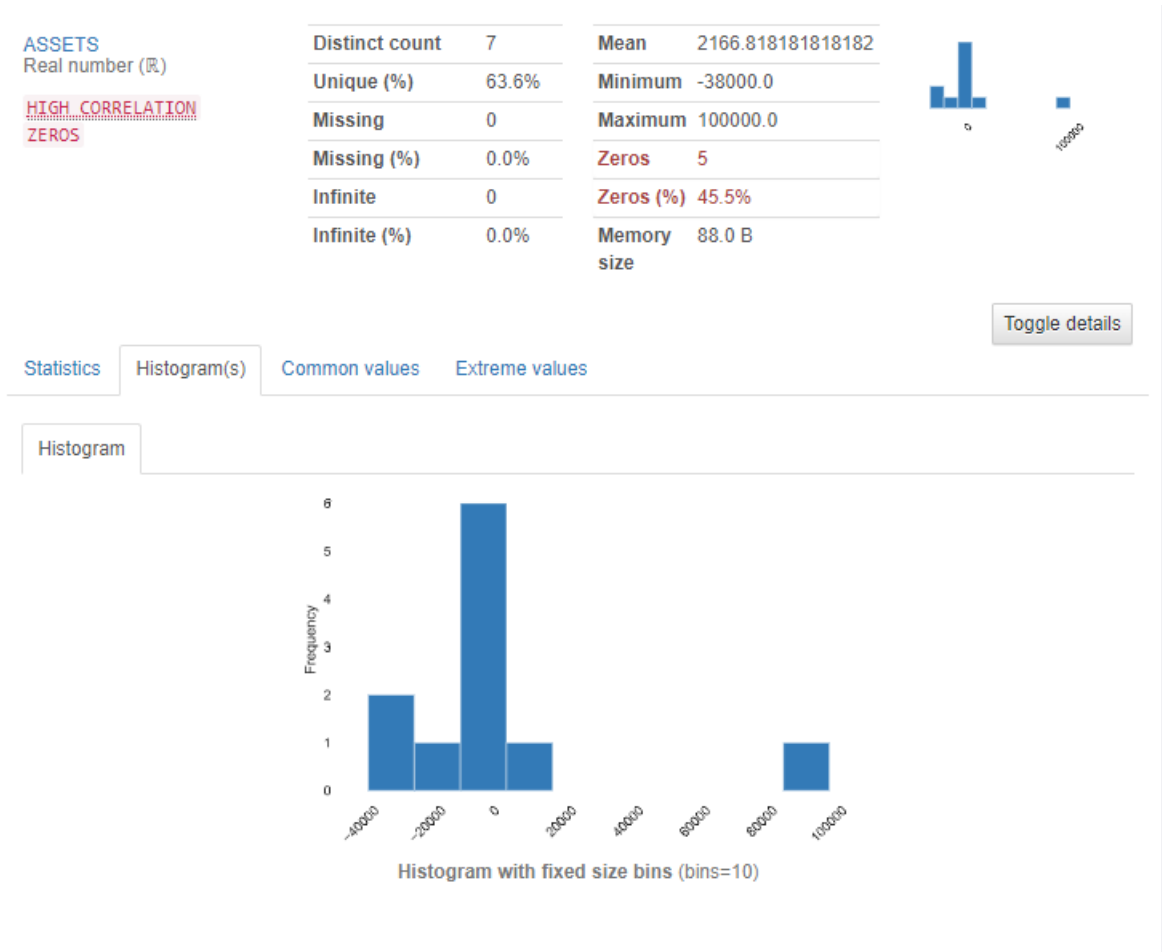
原贷款信息

贷款信息

贷款状态	贷款号	发放支行	查询/修改
发放中	L456456	上海支行	<input type="radio"/>
发放中	L567567	上海支行	<input type="radio"/>
已全部发放	L909089	上海支行	<input type="radio"/>
添加新贷款			<input type="radio"/>

4.2.4 业务统计

例：统计 2019/7/14 日后一年的银行财产规律



5 总结与讨论

本次实验中我收获了许多。无论是资料的查找、工具的选取，还是数据库与前端的设计、原理的理解，甚至是程序健壮性、可读性的考量，对我均是巨大的挑战。但是随着 10 天内不断查资料、问问题、思考，最终完成了这个大项目，使自己的学习能力、思考能力、逻辑能力、编程能力获得了较大的提升

学到的经验与教训：

1. 开发需要遵循一定的流程，自顶向下或自底向上均是不错的主意
2. 学会实行环境控制，保证开发环境中依赖的一致性，可以考虑使用 docker
3. 在大型应用中难以 debug 时，print 往往是一种比较有效的方式
4. 异常的捕获可以更准确地定位 bug 的所在，也能一定程度上处理未知错误、保证应用正常运行
5. 官方文档往往比较详细，遇到困难时可以去查 API 文档
6. 一个好的编辑器往往能提高编程效率
7. 学会功能封装、模块封装，每实现一个小功能建议立即 debug，防止代码过多难以定位