1. **前言**

笔者在3.31日老师布置的《银行储蓄业务-数据模型与功能设计》基础上进行了适当改动，进一步形成了银行储蓄业务系统的设计。

1. **业务描述**

一家银行有很多的储户（用户），每个用户可以开设多个账户。

储户的信息包括身份证号、姓名、性别、出生年月、电话、邮箱、邮编、地址等信息。

账户的信息包括账号、销户标记、账户余额等信息。

每条交易历史包括账户、存入/取出/还是转账、对方账号、摘要、金额、交易之前的余额、交易后的余额等信息。

一个账户对应很多的交易记录，记录了开户以来进行的历史交易。开户也需要登记一个历史记录。

需要注意的是，一般存入金额都是可行的。但是取钱，前提条件必须是账户里面有足够的钱，转账则需要转出的账户有足够的钱（余额）。

银行每年固定时间即某天，给银行账户计息。

计息规则是，根据上次计息以来，该账户在每个时间段的账户余额，进行计息，然后加起来，得到利息总额。一般是半夜计息。

比如，上次计息日为2020-1-1，之后到2020-1-2账户余额一直是10000元，后来有存入，到2020-1-5账户余额一直是20000元。2020-1-5计息。

那么，按照10000元持续1天，20000元持续3天计息，假设年利率为1.5%，那么两部分利息为，1\*10000\*1.5%/365，3 \*2000×1.5%/365，加起来作为这段时间的账户利息。

需要记录计息历史，包括账户、上次计息日、本次计息日、计息金额。

并且在交易历史里，记录计息操作，即存入计算出来的利息，摘要部分记录为“计息”。

系统管理员可以查看所有用户的账户数量、余额和银行总余额等。

1. **数据库设计**
2. 数据模型

经分析可知，实体及属性基本信息表如下：

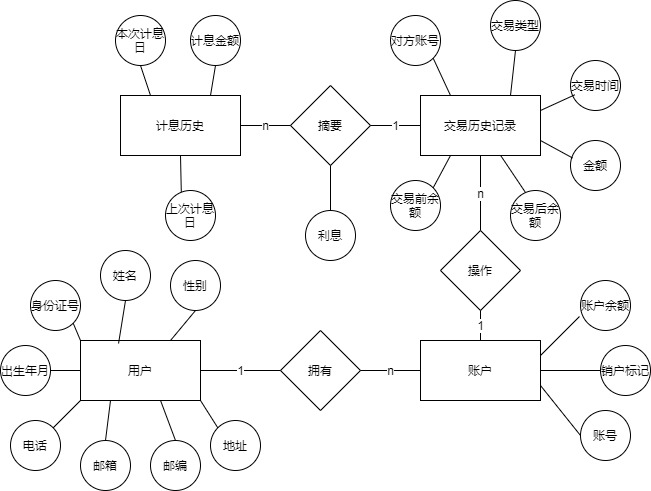
|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 基本属性 |
| 1 | 用户 | 身份证号、姓名、性别、出生年月、电话、邮箱、邮编、地址 |
| 2 | 账户 | 账号、销户标记、账户余额 |
| 3 | 交易历史记录 | 对方账号、交易金额、交易类型、交易前余额、交易后的余额、交易时间 |
| 4 | 计息历史 | 上次计息日、本次计息日、计息金额 |

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 相关实体 | 联系 | 联系属性 |
| 1 | 用户：账户 | 1 : n | 无 |
| 2 | 账户：交易历史记录 | 1 : n | 无 |
| 3 | 交易历史记录：计息历史 | 1 : n | 计息 |

对应的关系模型如下（主码标粗）：

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 属性 |
| 1 | 用户 | **身份证号**、姓名、性别、出生年月、电话、邮箱、邮编、地址 |
| 2 | 账户 | **账号**、销户标记、账户余额、储户身份证号 |
| 3 | 交易历史记录 | **账户、对方账号**、交易金额、交易类型、交易前余额、交易后的余额、**交易时间** |
| 4 | 计息历史 | 账户、上次计息日、**本次计息日**、计息金额 |

1. ER图



1. 源码（seed\_data.sql）

create database Bank1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

创建表

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--创建用户表

create table userlist(

name char(10) not null,--姓名

gender char(2),--性别

birthday date,--生日

ID\_number char(18),--身份证号码

tel\_number char(11),--电话号

email char(20),--邮箱

address char(30),--地址

primary key(ID\_number))

--创建账户表

create table accountlist(

account\_number char(6),--账号

cancelflag char(2) check(cancelflag in ('是','否')),--销户标记

balance float,--余额

user\_ID\_number char(18),--储户身份证号码

primary key(account\_number))

--创建交易历史记录表

create table trading\_record(

own\_account\_number char(6),--账号

opposite\_account\_number char(6),--对方账号

trade\_time datetime,--交易时间

trade\_type char(4) check(trade\_type in('存款','取款','转账','开户','计息')),

--交易时间

amount float,--交易金额

balance\_before float,--交易之前的余额

balance\_after float check(balance\_after>=0),--交易后的余额

primary key(own\_account\_number,trade\_time))

--创建计息历史表

create table interest\_history(

account\_number char(6),--账号

date\_last date,--上次计息日

date\_new date,--本次计息日

interest\_amount float,--计息金额

primary key(account\_number,date\_new))

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

插入数据

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

insert into userlist values('张一','男','2000-01-03','000000000000000001','00000000001','000001@163.com','上海市南京路');

insert into accountlist values('000001','否',0,'000000000000000001');

insert into trading\_record values('000001','000001','2020-01-01 00:00:00','开户',0,0,0)

insert into userlist values('王二','男','1987-12-22','000000000000000002','00000000002','000002@163.com','南京市上海路');

insert into accountlist values('000002','否',3000,'000000000000000002');

insert into accountlist values('000022','否',5000,'000000000000000002');

insert into trading\_record values('000002','000002','2013-04-01 00:00:00','开户',3000,0,3000)

insert into trading\_record values('000022','000022','2013-04-01 00:00:00','开户',5000,0,5000)

1. **模块设计及功能点描述**

使用者进入homepage页面,可以作为用户或管理员登录，或创建新的用户。

1. 创建新用户：

输入新用户的姓名、身份证号码等信息，存入用户表。

1. 用户
   1. 开户（open account）

输入新账户的账号和用户的身份证号码及开户时存入的金额，存入账户表，并记录在交易历史记录表。

* 1. 存钱（save money）

输入账号及存入金额，更新账户表和交易历史记录表。

* 1. 取钱（draw money）

输入账号及取出金额，更新账户表和交易历史记录表。

* 1. 转账（account transfer）

输入账号、对方账号及转账金额，更新账户表和交易历史记录表。

* 1. 查询交易记录（query trading record）

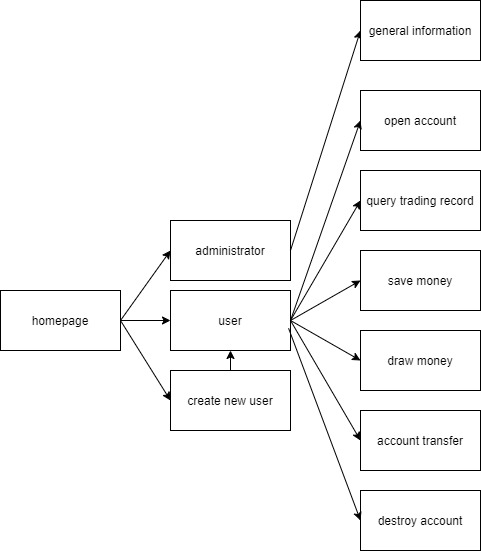
输入账号，查询交易历史记录表对应的内容。

* 1. 销户（destroy avvount）

输入账号，将账户的销户标记更新为‘是’，进行销户。

1. 管理员

查看所有用户的存款情况和银行总款等信息。



1. **系统实现描述**
   1. 建立数据库并插入数据(seed\_data.sql)；
   2. Python连接数据库，并返回网址以进行系统的操作。
      1. 开户
         1. 向账户中插入账户数据；
         2. 向交易历史记录中插入交易记录；
         3. 输出当前用户的所有账户；
      2. 销户
         1. 检查账户是否存在，如果不存在，则不能执行该操作；
         2. 将账户中销户标记更新为‘是’；
      3. 存钱
         1. 检查账户是否存在，如果不存在，则不能执行该操作；
         2. 查询账户余额；
         3. 向交易历史记录中插入数据；
         4. 更新账户实体中的账户余额；
         5. 输出存钱之前的余额和之后的余额；
      4. 取钱
2. 检查账户是否存在，如果不存在，则不能执行该操作；
3. 查询账户余额，如果取出金额大于余额，则不能执行该操作；
4. 向交易历史记录中插入数据；
5. 更新账户实体中的账户余额；
6. 输出存钱之前的余额和之后的余额；
   * 1. 转账
        1. 检查账户和对方账户是否存在，如果不存在，则不能执行该操作；
        2. 查询账户余额，如果转出金额大于余额，则不能执行该操作；
        3. 向交易历史记录中插入数据；
        4. 更新账户实体中的账户余额；
        5. 输出转账之前的余额和之后的余额；
     2. 计息
        1. 检查账户和对方账户是否存在，如果不存在，则不能执行该操作；
        2. 查询计息历史中的上次计息日；
        3. 查询交易历史记录中上次计息日以来各个时间段的账户余额；
        4. 计算利息总额，并向计息历史中插入数据；
        5. 向交易历史记录中插入数据；
        6. 更新账户中的账户余额；
     3. 查询总体情况
        1. 输出各用户的余额并用柱状图显示。
        2. 输出银行总余额。
7. **总结**

总的来说，本程序能够实现包括开户销户、存取钱、转账、查询交易记录等基本银行储蓄业务，输入数据对数据库进行动态的增删改查，并以表格和图片的形式在html上进行展示，笔者在编写程序时进一步加深了对SQL的理解。

另一方面，程序仍有许多不足和待改进之处，主要如下：

* + - 1. 功能模块不够完善。出于操作方便的考虑，笔者并没有在用户界面设置账号密码登录，而是直接输入账号便可进行业务处理，造成了系统的真实性和可用性大大降低。此外，银行系统应具有的贷款和计息功能并没有在程序中体现出来。
      2. 网页不够美观。由于笔者对web开发知识的欠缺，没能对网页做进一步的优化，没有将条形图和表格显示在同一页面上。

1. **参考文献**

[1]覃雄派老师示例代码 demo1.py

[2]《银行储蓄业务-数据模型与功能设计》