

“银行业务管理系统”

系统设计与实现报告

姓名：曾勇程

学号：PB18000268

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	2
2	总体设计	2
2.1	系统模块结构	2
2.2	系统工作流程	2
2.3	数据库设计	3
3	详细设计【可选】	9
3.1	登录模块	9
3.2	主菜单模块	9
3.3	客户管理模块	10
3.4	账户管理模块	11
3.5	贷款管理模块	15
3.3	业务统计模块	15
4	实现与测试	15
4.1	实现结果	15
4.2	测试结果	20
4.3	实现中的难点问题及解决【可选】	28
5	总结与讨论	28

1 概述

1.1 系统目标

本实验实现了一个银行数据库管理系统，该系统主要有 6 个功能，分别为支行管理，职工管理，客户管理，账户管理，贷款管理，统计查询等功能。该系统为每个功能实现了基本的条件查询、数据库插入，数据库删除和数据库更改等基本的数据库访问需求，针对每个功能还实现了简单美观的界面，很大程度地方便了用户进行数据管理。

1.2 需求说明

1. 数据库需求

银行业务管理系统的数据需求如下： 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帐户，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。

2. 功能需求

支行管理：提供支行所有信息的增、删、改、查功能；如果支行存在着关联信息，如员工、账户等，则不允许删除；

员工管理：提供支行员工所有信息的增、删、改、查功能；如果员工存在着关联数据，则不允许删除；

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

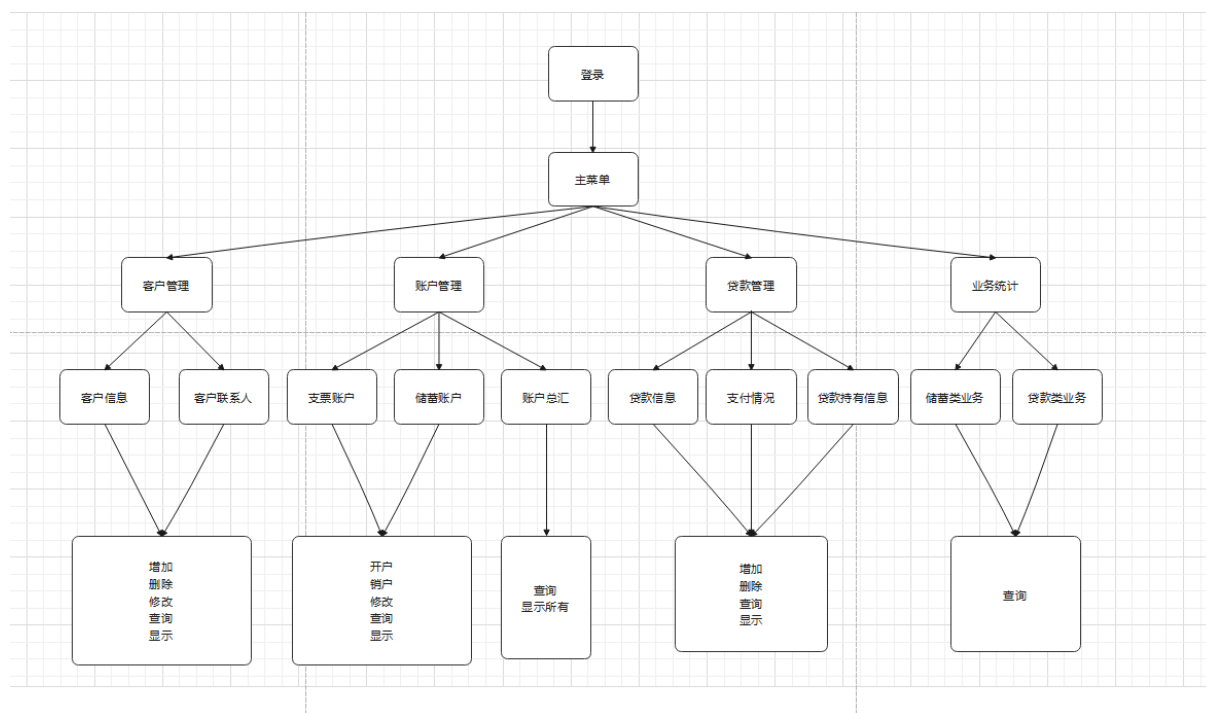
额和用户数。

1.2 本报告的主要贡献

1. 明确银行数据库管理系统的功能需求和基本的数据库需求
2. 总体概括系统的设计结构
3. 详细介绍系统的设计方法和程序流程
4. 对系统设计进行正确性测试和分析
5. 系统设计的总结与报告

2 总体设计

2.1 系统模块结构



2.2 系统工作流程

登录→选择需要使用的服务功能→退出系统
每次进入新的网页都会重连 MySQL

显示流程:

select_xx 获得对应的 query 语句 >> 调用对应的 get_query()函数访问数据库 >> 将数据显示在表格中

查询流程:

用户输入查询条件 >> 检查是否有不正规输入 >> 没有则 `select_xx` 获得对应的 `query` 语句 >> 调用对应的 `get_query()`函数访问数据库 >> 将数据显示在表格中

增加、开户流程:

用户输入查询条件 >> 检查是否有不正规输入 >> 没有则 `insert_xx` 获得对应的 `query` 语句 >> 调用对应的 `get_query()`函数访问数据库 >> 显示相应数据表格

删除、销户流程:

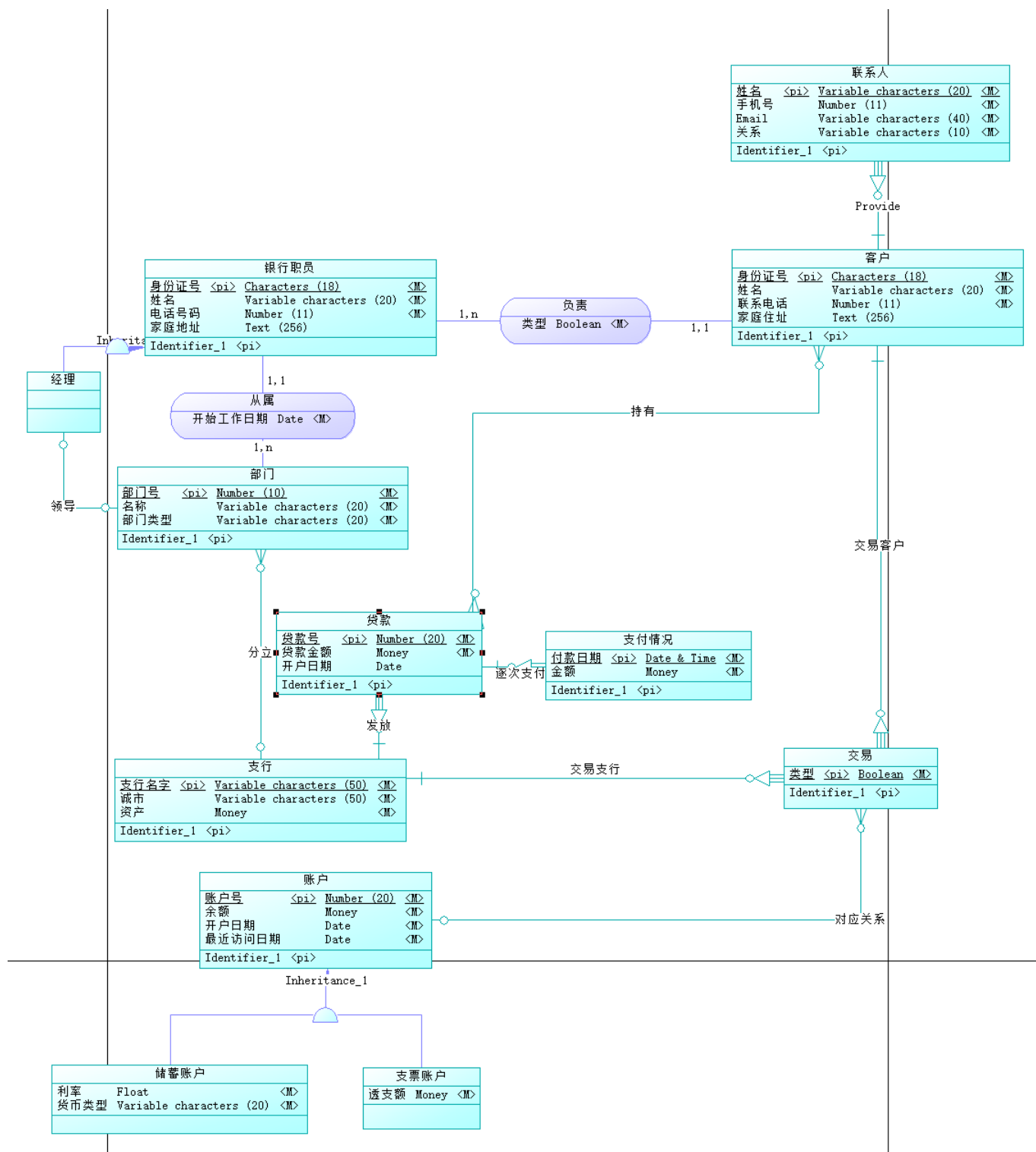
用户输入删除项目条件信息 >> 检查是否有不正规输入 >> 调用 `select_xx` 选中待删除项目并显示在表格中 >> 弹窗询问确定删除选中项目 >> 删除并显示最终数据结果

修改流程:

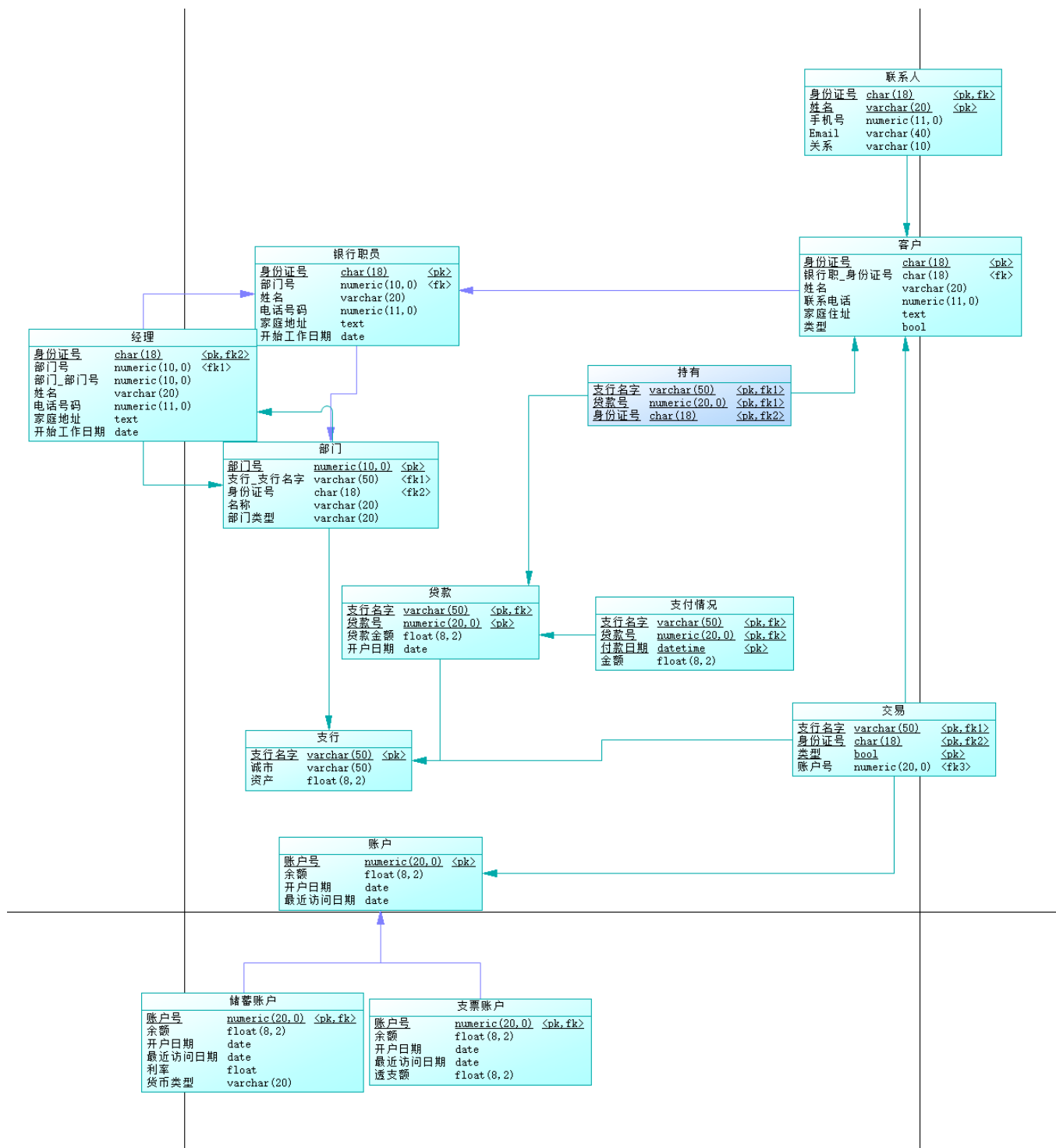
用户查询并选中要修改的项目 >> 在对应处输入修改后的值 >> 用户点击确认按钮 >> 系统检查是否有不正规输入 >> `update_xx` 获得对应的 `query` 语句 >> 弹窗询问是否确认写入 >> 确认写入则调用对应的 `get_query()`函数访问数据库 >> 显示修改后的结果

2.3 数据库设计

设计的 ER 图:



该 ER 图生成的物理模型:



生成的对应的数据库结构:

```

33  /*=====*/
34  /* Table: Account  账户                                */
35  /*=====*/
36  • create table Account
37  (
38      ID                numeric(20,0) not null,
39      balance           float(8,2) not null,
40      TIME_open         date not null,
41      TIME_access       date not null,
42      primary key (ID)
43  );
44
45  /*=====*/
46  /* Table: Bank  银行分行 (支行)                        */
47  /*=====*/
48  • create table Bank
49  (
50      name              varchar(50) not null,
51      city              varchar(50) not null,
52      property          float(8,2) not null,
53      primary key (name)
54  );
55
56  /*=====*/
57  /* Table: CK_account 支票账户                            */
58  /*=====*/
59  • create table CK_account
60  (
61      ID                numeric(20,0) not null,
62      balance           float(8,2) not null,
63      TIME_open         date not null,
64      TIME_access       date not null,
65      overdraft         float(8,2) not null,
66      primary key (ID)
67  );
68
69  /*=====*/
70  /* Table: Clerk  职员                                    */
71  /*=====*/
72  • create table Clerk
73  (
74      ID_card           char(18) not null,
75      ID_depert         numeric(10,0) not null,
76      name              varchar(20) not null,
77      telephone         numeric(11,0) not null,
78      address           text,
79      Data_Sw           date not null,
80      primary key (ID_card)
81  );

```



```

82
83      /*=====*/
84      /* Table: Client 客户 */
85      /*=====*/
86 • create table Client
87 (
88     ID_card          char(18) not null,
89     Cle_ID_card       char(18) not null,
90     name              varchar(20) not null,
91     telephone        numeric(11,0) not null,
92     address           text,
93     clerk_type        bool not null,
94     primary key (ID_card)
95 );
96
97      /*=====*/
98      /* Table: Contacts 联系人 */
99      /*=====*/
100 • create table Contacts
101 (
102     ID_card          char(18) not null,
103     name              varchar(20) not null,
104     telephone        numeric(11,0) not null,
105     Email             varchar(40) not null,
106     relation          varchar(10) not null,
107     primary key (ID_card, name)
108 );
109
110      /*=====*/
111      /* Table: Hold_C_L 持有信息 */
112      /*=====*/
113 • create table Hold_C_L
114 (
115     name              varchar(50) not null,
116     ID_loan           numeric(20,0) not null,
117     ID_card           char(18) not null,
118     primary key (name, ID_loan, ID_card)
119 );
120
121      /*=====*/
122      /* Table: Loans 贷款 */
123      /*=====*/
124 • create table Loans
125 (
126     name              varchar(50) not null,
127     ID_loan           numeric(20,0) not null,
128     money_loan        float(8,2) not null,
129     TIME_open         datetime not null,
130     primary key (name, ID_loan)
131 );

```

```
133      /*=====*/
134      /* Table: SV_account 储蓄账户 */
135      /*=====*/
136      • create table SV_account
137      (
138          ID                numeric(20,0) not null,
139          balance           float(8,2) not null,
140          TIME_open        date not null,
141          TIME_access       date not null,
142          rate              float not null,
143          currency          varchar(20) not null,
144          primary key (ID)
145      );
146
147      /*=====*/
148      /* Table: Transact 交易 */
149      /*=====*/
150      • create table Transact
151      (
152          name              varchar(50) not null,
153          ID_card           char(18) not null,
154          Type              bool not null,
155          ID                numeric(20,0),
156          primary key (name, ID_card, Type)
157      );
158
159      /*=====*/
160      /* Table: department 部门 */
161      /*=====*/
162      • create table department
163      (
164          ID_depert         numeric(10,0) not null,
165          Ban_name          varchar(50),
166          ID_card           char(18),
167          name              varchar(20) not null,
168          Type_depert       varchar(20) not null,
169          primary key (ID_depert)
170      );
171
172      /*=====*/
173      /* Table: payout 支付情况 */
174      /*=====*/
175      • create table payout
176      (
177          name              varchar(50) not null,
178          ID_loan           numeric(20,0) not null,
179          TIME_pay          datetime not null,
180          money_pay         float(8,2) not null,
181          primary key (name, ID_loan, TIME_pay)
182      );
183
184      /*=====*/
185      /* Table: ... */
186      /*=====*/
187      • create table ...
188      (
189          ...
190          primary key (...)
```

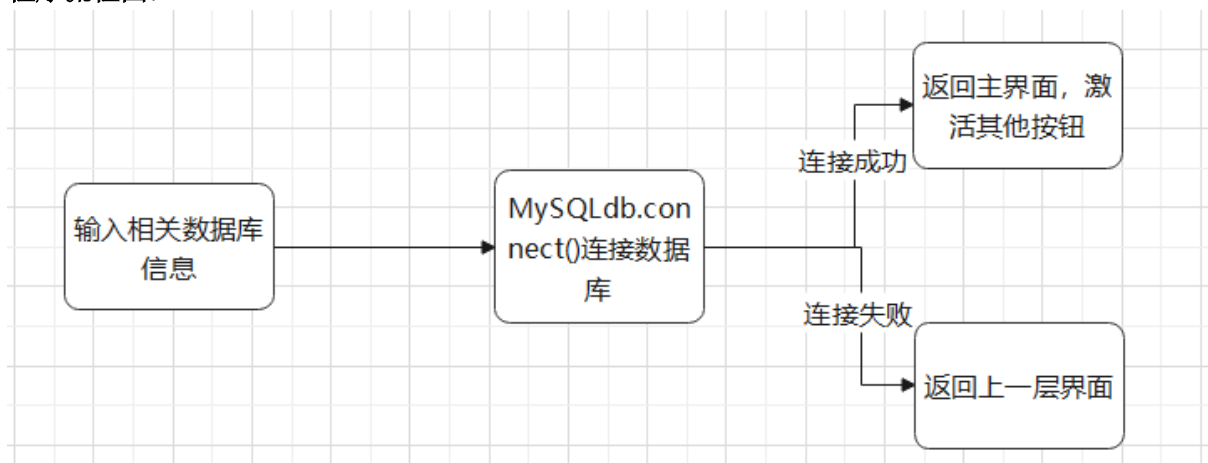
2 详细设计【可选】

3.1 登录模块

输入：数据库所在主机地址、访问的数据库名称、用户名、用户密码

输出：重定向至主菜单

程序流程图：



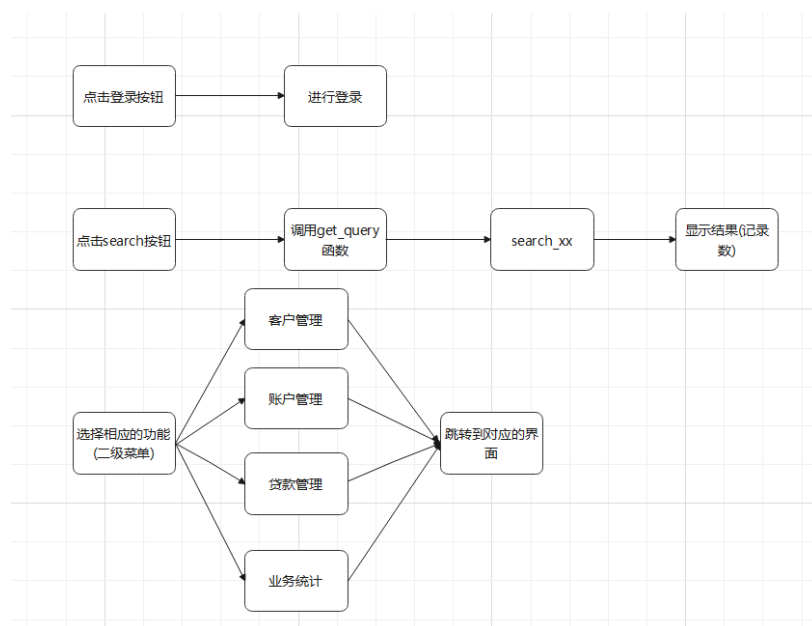
3.2 主菜单模块

输入：用户选择二级菜单按钮

输出：重定向至用户选择的二级菜单

其他功能：查看数据库内每个表的记录数目

程序流程图：



3.3 客户管理模块

输入：

客户信息界面：

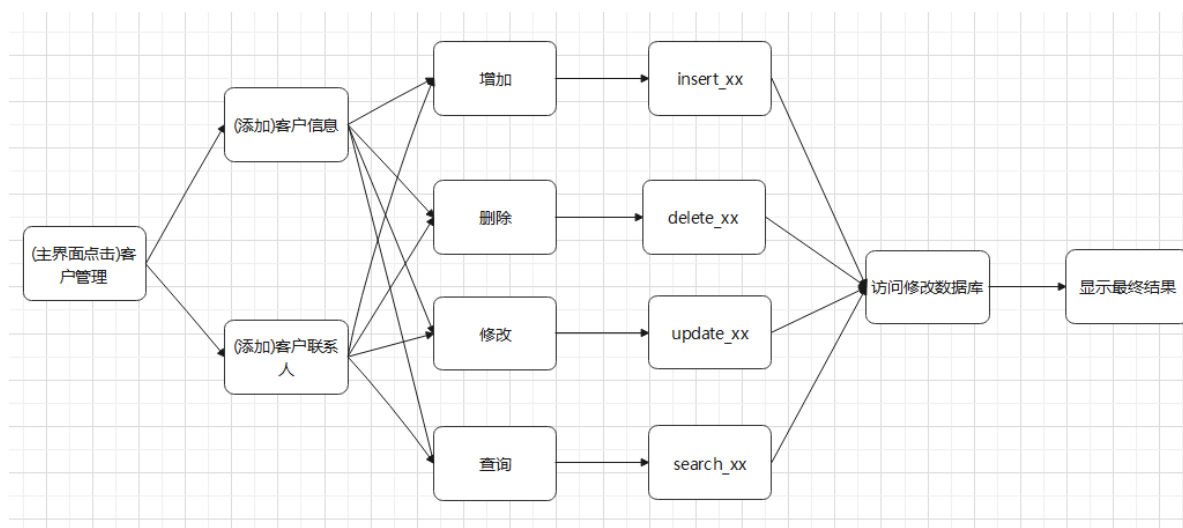
客户身份证号：	<input type="text"/>
银行职员身份证号：	<input type="text"/>
姓名：	<input type="text"/>
电话：	<input type="text"/>
家庭住址：	<input type="text"/>
银行职员类型：	<input type="text"/>
(0:贷款负责人, 1:银行账户负责人)	

客户联系人界面：

关联客户身份证：	<input type="text"/>
联系人姓名：	<input type="text"/>
联系人电话：	<input type="text"/>
联系人Email：	<input type="text"/>
关系：	<input type="text"/>

输出：显示最终操作完成后的结果

程序流程图：



3.4 账户管理模块

输入：

支票账户：

支票账户	
账户号：	<input type="text"/>
余额：	<input type="text"/>
开户日期：	<input type="text"/>
最近访问日期：	<input type="text"/>
透支额：	<input type="text"/>
支行名字：	<input type="text"/>
客户身份证号：	<input type="text"/>
日期格式如：2020.1.12 或 2020-01-12	

储蓄账户：

储蓄账户	
账户号：	<input type="text"/>
余额：	<input type="text"/>
开户日期：	<input type="text"/>
最近访问日期：	<input type="text"/>
利率：	<input type="text"/>
货币类型：	<input type="text"/>
支行名字：	<input type="text"/>
客户身份证号：	<input type="text"/>

日期格式如：2020.1.12 或 2020-01-12

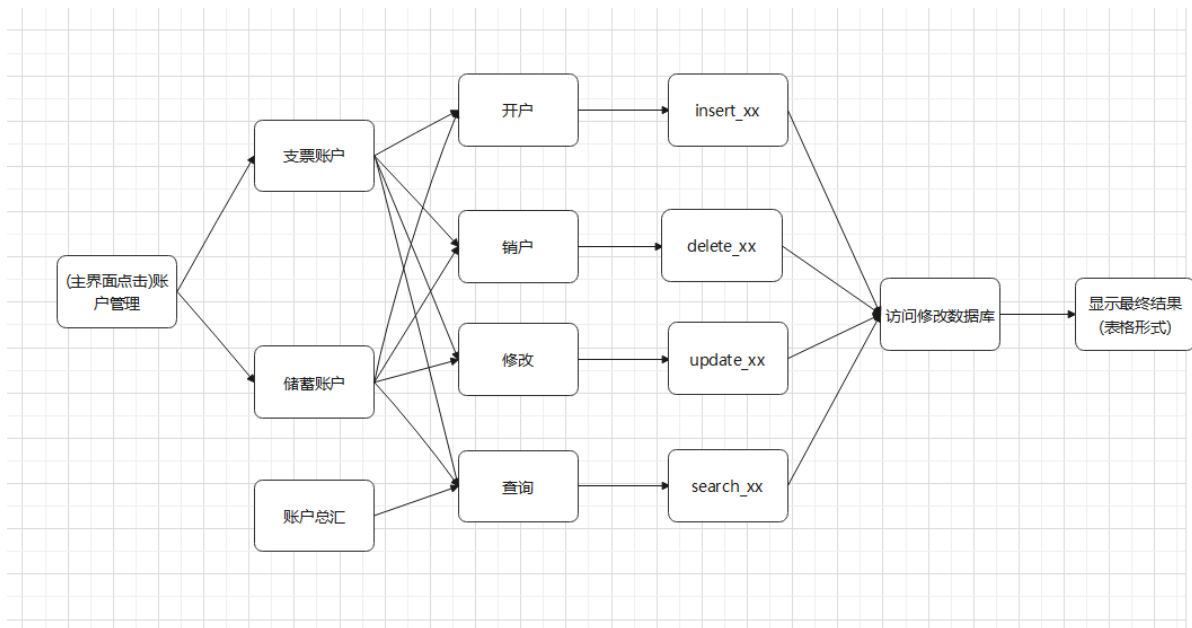
账户总汇：

账户总汇	
账户号：	<input type="text"/>
余额：	<input type="text"/>
开户日期：	<input type="text"/>
最近访问日期：	<input type="text"/>

日期格式如：2020.1.12 或 2020-01-12

输出：显示最终操作完成后的结果

程序流程图：



3.5 贷款管理模块

输入：

贷款信息：

贷款信息	
支行名字：	<input type="text"/>
贷款号：	<input type="text"/>
贷款金额：	<input type="text"/>
开户日期：	<input type="text"/>
客户身份证号：	<input type="text"/>

日期格式如：2020.1.12 或 2020-01-12

支付情况:

贷款信息

支行名字:

贷款号:

付款日期:

支付金额:

日期格式如: 2020.1.12 或 2020-01-12

贷款持有信息:

贷款所属信息

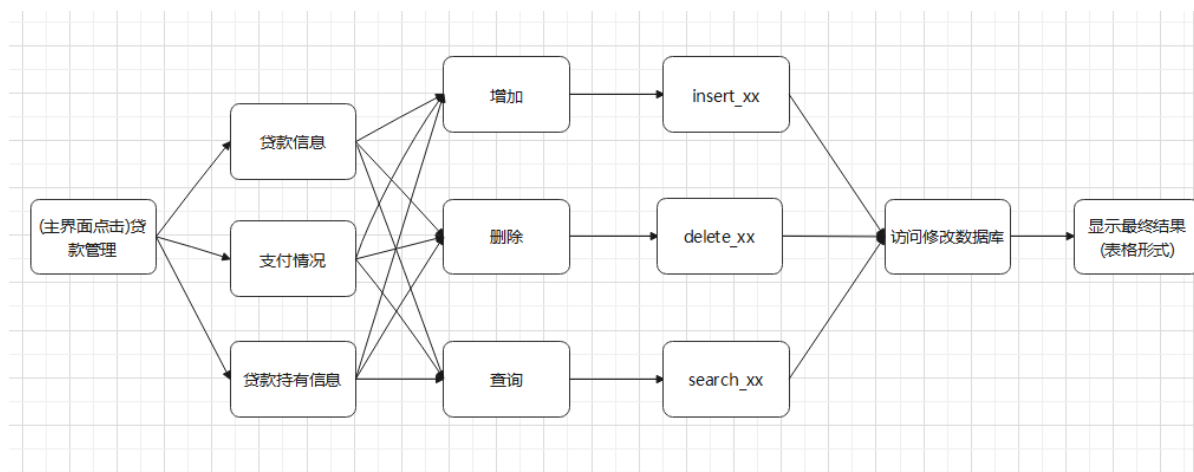
支行名字:

贷款号:

客户身份证号:

输出: 显示最终操作完成后的结果

程序流程图:



3.5 业务统计模块

输入：

储蓄类业务：

储蓄类业务统计

时间粒度： to

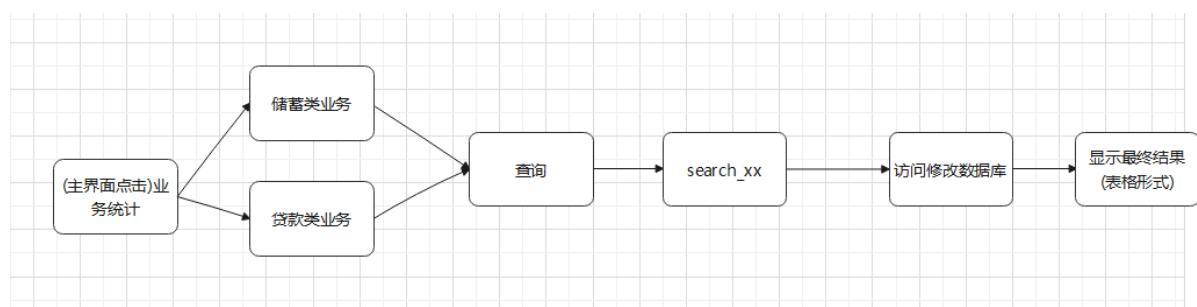
贷款类业务：

贷款类业务统计

时间粒度： to

输出：显示最终操作完成后的结果

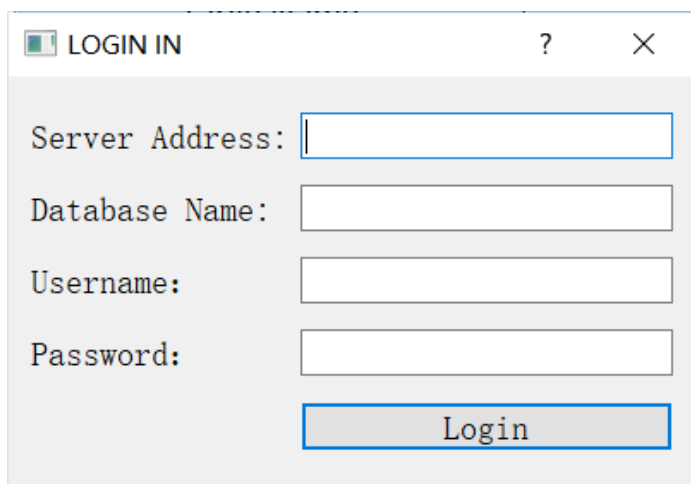
程序流程图：



4 实现与测试

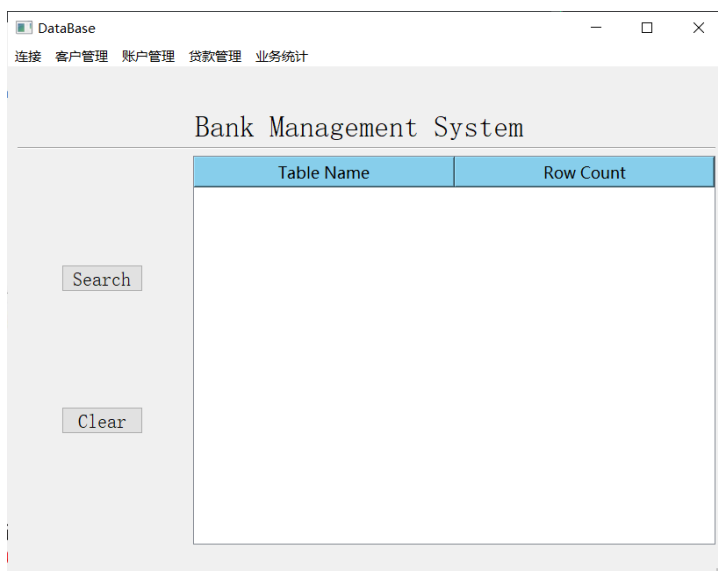
4.1 实现结果

登录界面：



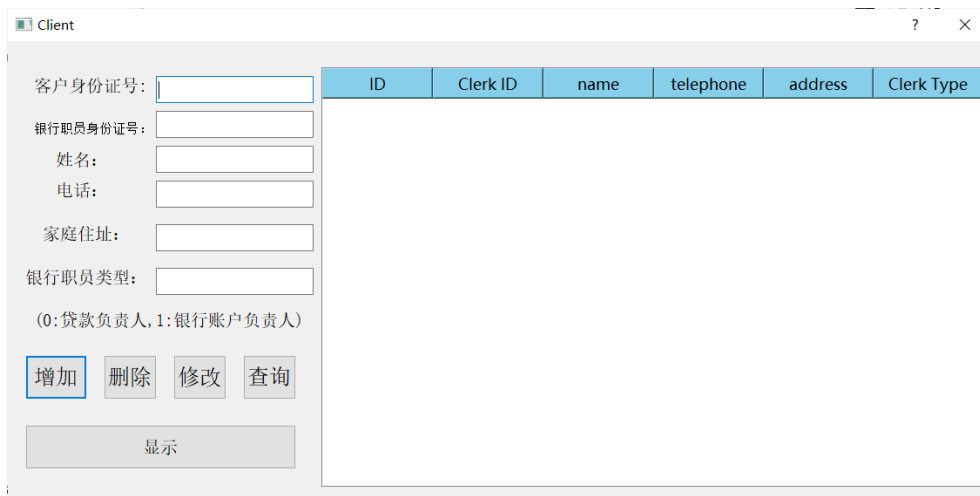
A dialog box titled "LOGIN IN" with a question mark icon and a close button. It contains four input fields: "Server Address:", "Database Name:", "Username:", and "Password:". Below these fields is a "Login" button.

主界面:



The main interface of the "Bank Management System". It has a title bar "DataBase" and a menu bar with "连接", "客户管理", "账户管理", "贷款管理", and "业务统计". The main area is titled "Bank Management System" and contains a table with two columns: "Table Name" and "Row Count". On the left side, there are "Search" and "Clear" buttons.

客户界面:



The "Client" interface. It has a title bar "Client" with a question mark icon and a close button. On the left, there are input fields for "客户身份证号:", "银行职员身份证号:", "姓名:", "电话:", "家庭住址:", and "银行职员类型:". Below these is a note "(0:贷款负责人, 1:银行账户负责人)". At the bottom left are buttons "增加", "删除", "修改", and "查询", and a "显示" button. On the right is a table with columns: "ID", "Clerk ID", "name", "telephone", "address", and "Clerk Type".

联系人界面：

Contact

?

×

关联客户身份证：

联系人姓名：

联系人电话：

联系人Email：

关系：

增加

删除

修改

查询

显示

ID_card	name	telephone	Email	relation
---------	------	-----------	-------	----------

支票账户界面：

CKAccount

?

×

支票账户

账户号：

余额：

开户日期：

最近访问日期：

透支额：

支行名字：

客户身份证号：

日期格式如：2020.1.12 或 2020-01-12

开户

销户

修改

查询

显示

ID	balance	TIME_open	TIME_access	overdraft	支行名字	所属客户身份证	账户类型
----	---------	-----------	-------------	-----------	------	---------	------

储蓄账户界面：

SVAccount

?

×

储蓄账户

账户号：

余额：

开户日期：

最近访问日期：

利率：

货币类型：

支行名字：

客户身份证号：

日期格式如：2020.1.12 或 2020-01-12

开户

销户

修改

查询

显示

ID	balance	TIME_open	TIME_access	rate	currency	支行名字	所属客户身份证	账户类型
----	---------	-----------	-------------	------	----------	------	---------	------

查询账户界面:

Account

?

×

账户总汇

账户号:

余额:

开户日期:

最近访问日期:

日期格式如: 2020.1.12 或 2020-01-12

查询

显示所有

ID	balance	TIME_open	TIME_access
----	---------	-----------	-------------

贷款界面:

Loan

?

×

贷款信息

支行名字:

贷款号:

贷款金额:

开户日期:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

增加

删除

查询

显示

支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
------	-----	------	------	----	--------

支付情况界面:

payout

?

×

贷款信息

支行名字:

贷款号:

付款日期:

支付金额:

日期格式如: 2020.1.12 或 2020-01-12

增加

删除

查询

显示

支行名字	贷款号	支付日期	支付金额
------	-----	------	------

贷款持有信息界面：

Hold

贷款所属信息

支行名字：

贷款号：

客户身份证号：

增加

删除

查询

显示

支行名字	贷款号	客户身份证号
------	-----	--------

储蓄类业务查询界面：

SVcount

储蓄类业务统计

时间粒度： to

查询

分行名称	用户数	业务总金额
------	-----	-------

贷款类业务查询界面：

Loancount

贷款类业务统计

时间粒度： to

查询

分行名称	用户数	贷款总金额
------	-----	-------

4.2 测试结果

1. 客户管理

身份证, 电话等存在位数要求, 银行职员类型存在值域(0 或 1)。

查询(按照填写信息进行查询, 未填则显示所有):

The screenshot shows the 'Client' application window. On the left is a search form with the following fields: '客户身份证号:' (Customer ID Card No.) with value '330881200101248318', '银行职员身份证号:' (Bank Clerk ID Card No.), '姓名:' (Name), '电话:' (Phone), '家庭住址:' (Home Address), and '银行职员类型:' (Bank Clerk Type) with a dropdown menu. Below these fields are buttons for '增加' (Add), '删除' (Delete), '修改' (Modify), and '查询' (Query), and a '显示' (Display) button. On the right is a table with columns: ID, Clerk ID, name, telephone, address, and Clerk Type. The table contains one record with ID 1, Clerk ID 330881200..., name ZYC, telephone 178551030..., address 合肥 (Hefei), and Clerk Type 1.

ID	Clerk ID	name	telephone	address	Clerk Type
1	330881200...	ZYC	178551030...	合肥	1

显示 (显示所有信息, 由图中可以看出, 该数据库具有一定的识别错误能力, 如能正确显示名字: “O’ Neil”):

The screenshot shows the 'Client' application window with the same search form as before. The '显示' (Display) button has been clicked, and the table now displays 11 records. The records are as follows:

ID	Clerk ID	name	telephone	address	Clerk Type
1	111111115...	ffd	15518234...	beijing	1
2	111134765...	O' Neil	18797686...	America	1
3	222222354...	wtr	10023945...	beijing	1
4	330881200...	ZYC	17855103...	合肥	1
5	333346534...	hdjs	13346964...	heifei	0
6	444444534...	ejiod	13456788...	guangzhou	0
7	555523568...	jddi	15567676...	hangzhou	0
8	666666676...	jiid	16868754...	nanjing	1
9	777777468...	dhih	17764775...	beijing	1
10	888878765...	doid	17898654...	hefei	0
11	999998765...	djoh	18944556...	hefei	1

增加:

Client

客户身份证号: 121234567865345678

银行职员身份证号: 111111111111111111

姓名: 阿呆

电话: 17853456452

家庭住址: 内蒙古

银行职员类型: 0
(0: 贷款负责人, 1: 银行账户负责人)

增加 删除 修改 查询

显示

	ID	Clerk ID	name	telephone	address	Clerk Type
1	111111115...	11111111...	ffd	15518234...	beijing	1
2	111134765...	11111111...	O' Neil	18797686...	America	1
3	121234567...	11111111...	阿呆	17853456...	内蒙古	0
4	222222354...	22222222...	wtr	10023945...	beijing	1
5	330881200...	11111111...	ZYC	17855103...	合肥	1
6	333346534...	33333333...	hdjs	13346964...	heifei	0
7	444444534...	44444444...	ejiod	13456788...	guangzhou	0
8	555523568...	55555555...	jddi	15567676...	hangzhou	0
9	666666676...	66666666...	jiid	16868754...	nanjing	1
10	777777468...	77777777...	dhih	17764775...	beijing	1
11	888878765...	88888888...	doid	17898654...	hefei	0
12	999998765...	99999999...	djoh	18944556...	hefei	1

修改:

修改界面:

New Value

修改前 修改后

客户身份证号:

银行职员身份证号:

姓名:

电话:

家庭住址:

银行职员类型:

(0: 贷款负责人, 1: 银行账户负责人)

确定

修改测试:

New Value

修改前 修改后

客户身份证号: 121234567865345678

银行职员身份证号:

姓名: 小花

电话:

家庭住址:

银行职员类型:

(0: 贷款负责人, 1: 银行账户负责人)

确定

Client

客户身份证号: 121234567865345678

银行职员身份证号: 111111111111111111

姓名: 阿呆

电话: 17853456452

家庭住址: 内蒙古

银行职员类型: 0

(0: 贷款负责人, 1: 银行账户负责人)

增加 删除 修改 查询

显示

	ID	Clerk ID	name	telephone	address	Clerk Type
1	111111115...	11111111...	ffd	15518234...	beijing	1
2	111134765...	11111111...	O' Neil	18797686...	America	1
3	121234567...	11111111...	小花	17853456...	内蒙古	0
4	222222354...	22222222...	wtr	10023945...	beijing	1
5	330881200...	11111111...	ZYC	17855103...	合肥	1
6	333346534...	33333333...	hdjs	13346964...	heifei	0
7	444444534...	44444444...	ejiod	13456788...	guangzhou	0
8	555523568...	55555555...	jddi	15567676...	hangzhou	0
9	666666676...	66666666...	jiid	16868754...	nanjing	1
10	777777468...	77777777...	dhih	17764775...	beijing	1
11	888878765...	88888888...	doid	17898654...	hefei	0
12	999998765...	99999999...	djoh	18944556...	hefei	1

删除(删除小花信息):

Client

客户身份证号: 121234567865345678

银行职员身份证号:

姓名:

电话:

家庭住址:

银行职员类型:

(0: 贷款负责人, 1: 银行账户负责人)

增加 删除 修改 查询

显示

	ID	Clerk ID	name	telephone	address	Clerk Type
1	111111115...	11111111...	ffd	15518234...	beijing	1
2	111134765...	11111111...	O' Neil	18797686...	America	1
3	222222354...	22222222...	wtr	10023945...	beijing	1
4	330881200...	11111111...	ZYC	17855103...	合肥	1
5	333346534...	33333333...	hdjs	13346964...	heifei	0
6	444444534...	44444444...	ejiod	13456788...	guangzhou	0
7	555523568...	55555555...	jddi	15567676...	hangzhou	0
8	666666676...	66666666...	jiid	16868754...	nanjing	1
9	777777468...	77777777...	dhih	17764775...	beijing	1
10	888878765...	88888888...	doid	17898654...	hefei	0
11	999998765...	99999999...	djoh	18944556...	hefei	1

客户的联系人的增删改查方式和实现方式和上面的客户管理十分类似, 就不再重复叙述。

2. 账户管理:

能够自动识别日期格式是否正确, 对于不正确的日期格式, 会报错并返回。

支票账户:

开户:

支票账户

账户号:

余额:

开户日期:

最近访问日期:

透支额:

支行名字:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

ID	balance	TIME_open	TIME_access	overdraft	支行名字	所属客户身份	账户类型
1001	1000.0	2001-01-01	2020-12-01	10.0	中商分行1	33088120...	0

查询:

支票账户

账户号:

余额:

开户日期:

最近访问日期:

透支额:

支行名字:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

ID	balance	TIME_open	TIME_access	overdraft	支行名字	所属客户身份	账户类型
1001	1000.0	2001-01-01	2020-12-01	10.0	中商分行1	33088120...	0

修改:

New Value

修改前 修改后

账户号: 账户号不得修改

余额:

开户日期:

最近访问日期:

透支额:

支行名字:

客户ID:

日期格式如: 2020.1.12 或 2020-01-12

支票账户

账户号:

余额:

开户日期:

最近访问日期:

透支额:

支行名字:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

开户 销户 修改 查询

显示

ID	balance	TIME_open	TIME_access	overdraft	支行名字	所属客户身份	账户类型
1001	100000.0	2001-01-01	2020-12-01	10.0	中商分行1	33088120...	0

销户(支票账户中只有唯一的一条记录, 被删除后数据库为空):

支票账户

账户号:

余额:

开户日期:

最近访问日期:

透支额:

支行名字:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

开户 销户 修改 查询

显示

ID	balance	TIME_open	TIME_access	overdraft	支行名字	所属客户身份	账户类型
----	---------	-----------	-------------	-----------	------	--------	------

储蓄账户的增删改查方式和实现方式和上面的支票账户十分类似, 就不再赘述。
账户总汇只有查询功能, 实现方式与支票账户类似, 同样不再赘述。

3. 贷款管理:

显示:

Loan

贷款信息

支行名字:

贷款号:

贷款金额:

开户日期:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

增加 删除 查询

显示

	支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
1	中商分行1	1034	1000.0	2001-10-0...	发放中	11111111...
2	中商分行2	1044	1000.0	2001-10-0...	发放中	11111111...
3	中商分行3	1035	1000.0	2001-10-0...	未开始发放	11111111...
4	中商分行7	1011	1000.0	2019-10-0...	未开始发放	11111111...
5	中商分行8	1011	1000.0	2010-10-0...	未开始发放	11111111...
6	中商分行9	1011	1000.0	1995-10-0...	已全部发放	11111111...

payout ? X

贷款信息

支行名字:

贷款号:

付款日期:

支付金额:

日期格式如: 2020.1.12 或 2020-01-12

增加 删除 查询

显示

	支行名字	贷款号	支付日期	支付金额
1	中商分行1	1034	2002-01-...	500.0
2	中商分行2	1044	2002-01-...	500.0
3	中商分行9	1011	2001-01-...	500.0
4	中商分行9	1011	2002-01-...	500.0

增加:

Loan ? X

贷款信息

支行名字:

贷款号:

贷款金额:

开户日期:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

增加 删除 查询

显示

	支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
1	中商分行1	1034	1000.0	2001-10-0...	发放中	11111111...
2	中商分行2	1044	1000.0	2001-10-0...	发放中	11111111...
3	中商分行3	1035	1000.0	2001-10-0...	未开始发放	11111111...
4	中商分行7	1011	1000.0	2019-10-0...	未开始发放	11111111...
5	中商分行8	1011	1000.0	2010-10-0...	未开始发放	11111111...
6	中商分行9	1011	1000.0	1995-10-0...	已全部发放	11111111...
7	中商分行4	1067	5000.0	2006-01-0...	未开始发放	33088120...

查询:

Loan ? X

贷款信息

支行名字:

贷款号:

贷款金额:

开户日期:

客户身份证号:

日期格式如: 2020.1.12 或 2020-01-12

增加 删除 查询

显示

	支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
1	中商分行4	1067	5000.0	2006-01-0...	未开始发放	33088120...

删除（删除匹配条件的信息）：

Loan

?

×

贷款信息

支行名字：

贷款号：

贷款金额：

开户日期：

客户身份证号：

日期格式如：2020.1.12 或 2020-01-12

增加

删除

查询

显示

	支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
1	中商分行1	1034	1000.0	2001-10-0...	发放中	11111111...
2	中商分行2	1044	1000.0	2001-10-0...	发放中	11111111...
3	中商分行3	1035	1000.0	2001-10-0...	未开始发放	11111111...
4	中商分行7	1011	1000.0	2019-10-0...	未开始发放	11111111...
5	中商分行8	1011	1000.0	2010-10-0...	未开始发放	11111111...
6	中商分行9	1011	1000.0	1995-10-0...	已全部发放	11111111...

若删除“发放中”的贷款，则会报错：

Loan

?

×

贷款信息

支行名字：

贷款号：

贷款金额：

开户日期：

客户身份证号：

日期格式如：2020.1.12 或 2020-01-12

增加

删除

查询

显示

	支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
1	中商分行1	1034	1000.0	2001-10-0...	发放中	11111111...
2	中商分行2	1044	1000.0	2001-10-0...	发放中	11111111...
3	中商分行3	1035	1000.0	2001-10-0...	未开始发放	11111111...
4	中商			2019-10-0...	未开始发放	11111111...
5	中商			2010-10-0...	未开始发放	11111111...
6	中商			1995-10-0...	已全部发放	11111111...

消息

×

此贷款正在发放中,不允许删除!

Yes

No

但是可以删除“已全部发放”的贷款：

Loan

?

×

贷款信息

支行名字：

贷款号：

贷款金额：

开户日期：

客户身份证号：

日期格式如：2020.1.12 或 2020-01-12

增加

删除

查询

显示

	支行名字	贷款号	贷款金额	开户日期	状态	客户身份证号
1	中商分行1	1034	1000.0	2001-10-0...	发放中	11111111...
2	中商分行2	1044	1000.0	2001-10-0...	发放中	11111111...
3	中商分行3	1035	1000.0	2001-10-0...	未开始发放	11111111...
4	中商分行7	1011	1000.0	2019-10-0...	未开始发放	11111111...
5	中商分行8	1011	1000.0	2010-10-0...	未开始发放	11111111...

此时，会将支付情况里的支付记录一并（原子地）删除：

payout

贷款信息

支行名字:

贷款号:

付款日期:

支付金额:

日期格式如: 2020.1.12 或 2020-01-12

增加 删除 查询

显示

	支行名字	贷款号	支付日期	支付金额
1	中商分行1	1034	2002-01-...	500.0
2	中商分行2	1044	2002-01-...	500.0

贷款持有信息界面用于给已有的贷款添加其他持有客户或删除多余客户（一个贷款可以被多个用户共同占有），实现方式和上面的类似，但比上面简单，因此也不再赘述。

4. 业务统计:

储蓄类业务统计:

SVcount

储蓄类业务统计

时间粒度: 1980/1/1 to 2021/1/1

查询

	分行名称	用户数	业务总金额
1	中商分行1	1	1000.0
2	中商分行5	1	1000.0
3	中商分行3	1	1000.0
4	中商分行9	1	1000.0

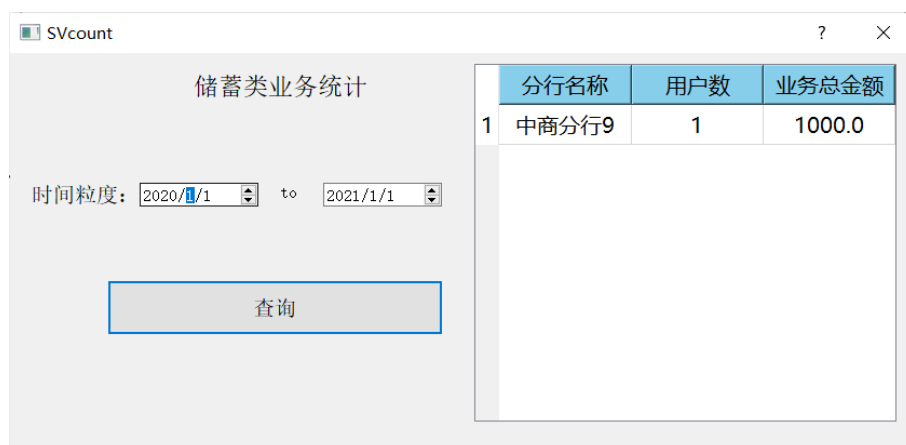
SVcount

储蓄类业务统计

时间粒度: 2006/1/1 to 2021/1/1

查询

	分行名称	用户数	业务总金额
1	中商分行5	1	1000.0
2	中商分行3	1	1000.0
3	中商分行9	1	1000.0



	分行名称	用户数	业务总金额
1	中商分行9	1	1000.0

贷款类业务与储蓄类业务的界面相似，实现方法也类似，因此不再赘述。

由上面的测试结果可以看出，数据库的所有要求的功能都已经实现，且同时也较好地实现了数据库的 ACID 性质。

4.3 实现中的难点问题及解决【可选】

其中的一个难点是由于关系模式很多，代码量随着编程的推进，变得冗长且不易调试，解决方法是将每个关系模式归为一个模块，进行模块化编程，这样，虽然总体代码量没有减少，但每个文件的代码量大大减少，并且也易于调试。

另一个难点在于每个功能模块的设计类似，但由于编程规模较大，复制粘贴然后修改特定的值很花时间，解决方法为将比较常用的每个功能模块都会用到的变量宏定义，赋值到一个 `self` 的变量中，这样就只需要修改几处就能完成一个任务模块的编写，十分便捷且必要。

5 总结与讨论

数据库的实现过程大致要注重以下几个方向：

1. 分析好需求，按需建立数据库和插入数据库条目。
2. 保证数据库的 ACID 性质，当出现需求分析不足时需要修改数据库设计。
3. 界面设计尽量方便用户使用习惯，避免对用户造成不便和不适，并针对一些特殊类型给出正确的输入格式的例子，如日期的正确输入格式。
4. 开发和测试可以同时进行，首先保证单个界面的逻辑通畅，然后迁移到其他的逻辑界面上去。
5. 设计一个功能时最好统一接口，规范化代码，这样设计其他功能时，能很方便地借鉴第一个功能的设计，并且也方便出错时调试。
6. 必须预测一些输入时可能出现的错误，并在编程时加以预防，以提高数据库的安全性和稳定性，不会轻易死机。