

# 银行业务管理系统数据库设计

学号 PB20061338 姓名 柯志伟

## 1 概念模型设计

### 1.1 实体设计

实体	实体属性	实体标识	设计理由
Bank	B_Name、City、Assets	B_Name	银行的每个支行具有自身的一些属性并于其他实体发生联系
Client	C_ID、C_Name、C_Tel、C_Addr	C_ID	客户具有多个属性，且会与员工、银行等实体发生联系
Contact	Co_ID、Co_Name、Co_Tel、Co_Email	E_ID	
Employee	E_ID、E_Name、E_Tel、E_Addr、E_Date	E_ID	每个支行的员工具有多个属性，并与客户、部门等发生联系
Department	D_ID、D_Name、D_Type、Manager_ID	D_ID	部门包含了多个属性，且每个部门可能含有多个员工，每个支行可能含有多个部门
Account	A_ID、Balance、StartDate	A_ID	账户具有自身的一些属性，且分成两个子类，与支行和客户发生联系
SavingAccount	InterestRate、SA_Type		作为账户实体的子类，具有自身特有的属性,利率和货币类型
CheckAccount	OverDraft		作为账户实体的子类，具有自身的属性，透支额
Loan	L_ID、L_Money、Count	L_ID	贷款具有多个属性，且与银行和客户间存在联系
Pay	P_ID、PayDate、PayMoney	P_ID	支行针对自身借贷出的每个贷款可能分多次支付,且具有支付金额，第几次支付等属性

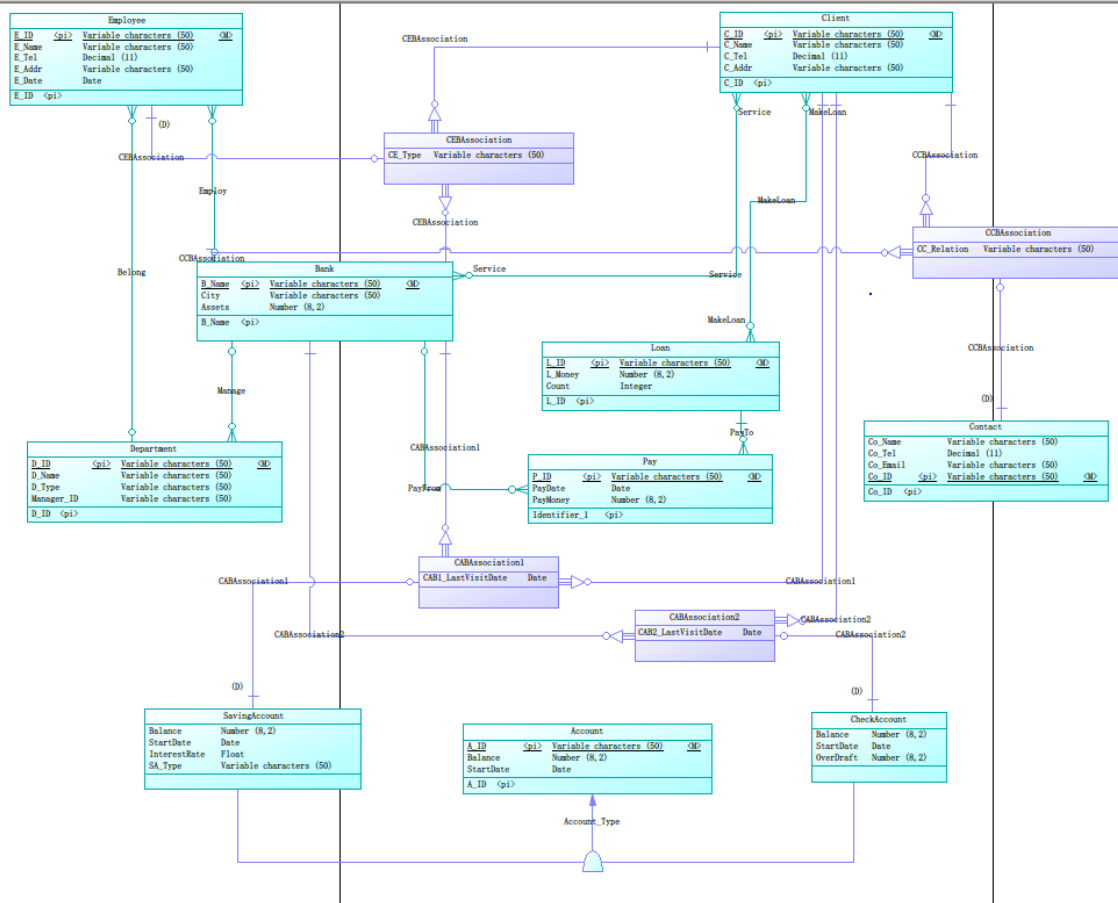
对应含义如下:

实体	实体属性	实体标识	设计理由
支行	名字、城市、资产	名字	见上
客户	身份证号、名字、联系电话、家庭住址	身份证号	见上
联系人	身份证号、姓名、手机号、Email	身份证号	见上
员工	身份证号、姓名、电话号码、家庭住址、开始工作日期	身份证号	见上
部门	部门号、部门名称、部门类型、部门经理身份证号	部门号	见上
账户	账户号、余额、开户日期	账户号	见上
储蓄账户	利率、货币类型		见上
支票账户	透支额		见上
贷款	贷款号、贷款余额、逐次支付情况	贷款号	见上
付款	付款号、付款日期、付款金额	付款号	见上

## 1.2 联系设计

联系	联系实体	联系属性	设计理由
Service	Client、 Bank	无	银行为客户提供服务、且两者之间关系的约束为 <b>客户:银行=多:多</b>
Employ	Bank、 Employee	无	银行雇佣职员、且两者之间关系的约束为 <b>银行:职工=1:多</b>
Belong	Employee、 Department	无	每个银行职员隶属银行的某个部门、且两者之间关系的约束为 <b>职工:部门=多:1</b>
Manage	Bank、 Department	无	银行管理多个部门、两者之间关系的约束为 <b>银行:部门=1:多</b>
MakeLoan	Client、 Loan	无	客户会申请贷款、且一个贷款可能为多人所有、两者之间关系的约束为 <b>客户:贷款=多:多</b>
PayFrom	Pay、 Bank	无	银行向贷款支付、且可能分多次支付、两者之间关系的约束为 <b>付款:银行=多:1</b>
PayTo	Pay、 Loan	无	每次付款都是针对一个贷款的、两者之间关系的约束为 <b>付款:贷款=多:1</b>
CEBAssociation	Bank、 Client、 Employee	CE_Type	银行要求每个客户与一个银行员工发生联系且存储两者的关系,三者之前关系的约束是 <b>银行:客户:职工=多:多:1</b>
CCBAssociation	Client、 Contact、 Bank	CC_Relation	支行要求存有它的客户的一位联系人信息,联系人和客户之间的关系是一对多,两者之间关系的约束为 <b>客户:银行:联系人=多:多:1</b>
CABAssociation1	Bank、 SavingAccount、 Client	CAB1_LastVisitDate	每个储蓄账户与某个银行关联并属于某个客户,每个支行要求每个客户只能有一个储蓄账号,三者之前关系的约束是 <b>银行:储蓄账号:客户=多:1:多</b>
CABAssociation2	Bank、 CheckAccount、 Client	CAB2_LastVisitDate	每个支票账户与某个银行关联并属于某个客户,每个支行要求每个客户只能有一个支票账号,三者之前关系的约束是 <b>银行:支票账号:客户=多:1:多</b>

## 1.3 Power Designer的ER图



## 2 概念模型到逻辑模型的转换

### 2.1 实体转换

将实体转换如下：

Bank (**B\_Name**、City、Assets)

Client (**C\_ID**、C\_Name、C\_Tel、C\_Addr)

Contact (**Co\_ID**、Co\_Name、Co\_Tel、Co\_Email)

Employee (**E\_ID**、E\_Name、E\_Tel、E\_Addr、E\_Date)

Department (**D\_ID**、D\_Name、D\_Type、Manager\_ID)

Account (**A\_ID**、Balance、StartDate)

SavingAccount (**A\_ID**、InterestRate、SA\_Type)

CheckAccount (**A\_ID**、OverDraft)

Loan (**L\_ID**、L\_Money、Count)

Pay (**P\_ID**、PayDate、PayMoney)

### 2.2 联系转换

- **Service** 关系的约束为 **客户:银行=多:多**

Service (**C\_ID**、**B\_Name**)

- **Employ** 关系的约束为 **银行:职工=1:多**

Employee (**E\_ID**、E\_Name、E\_Tel、E\_Addr、E\_Date、**B\_Name**)

- **Belong** 关系的约束为 **职工:部门=多:1**  
Employee (**E\_ID**、E\_Name、E\_Tel、E\_Addr、E\_Date、B\_Name、D\_ID)
- **Manage** 关系的约束为 **银行:部门=1:多**  
Department (**D\_ID**、D\_Name、D\_Type、Manager\_ID、B\_Name)
- **MakeLoan** 关系的约束为 **客户:贷款=多:多**  
MakeLoan (**C\_ID**、**L\_ID**)
- **PayFrom** 关系的约束为 **付款:银行=多:1**  
Pay (**P\_ID**、PayDate、PayMoney、B\_Name)
- **PayTo** 关系的约束为 **付款:贷款=多:1**  
Pay (**P\_ID**、PayDate、PayMoney、B\_Name、L\_ID)
- **CEAssociation** 关系的约束是 **银行:客户:职工=多:多:1**  
CEBAssociation (**B\_Name**、**C\_ID**、E\_ID)
- **CCBAssociation** 关系的约束为 **客户:银行:联系人=多:多:1**  
CCBAssociation (**C\_ID**、**B\_Name**、Co\_ID)
- **CABAssociation1** 关系的约束是 **银行:储蓄账号:客户=多:1:多**  
CABAssociation1 (**B\_Name**、**C\_ID**、A\_ID)
- **CABAssociation2** 关系的约束是 **银行:支票账号:客户=多:1:多**  
CABAssociation2 (**B\_Name**、**C\_ID**、A\_ID)

## 2.3 最终的关系模式

Bank (**B\_Name**、City、Assets)

Client (**C\_ID**、C\_Name、C\_Tel、C\_Addr)

Contact (**Co\_ID**、Co\_Name、Co\_Tel、Co\_Email)

Employee (**E\_ID**、E\_Name、E\_Tel、E\_Addr、E\_Date、B\_Name、D\_ID)

Department (**D\_ID**、D\_Name、D\_Type、Manager\_ID、B\_Name)

Account (**A\_ID**、Balance、StartDate)

SavingAccount (**A\_ID**、InterestRate、SA\_Type)

CheckAccount (**A\_ID**、OverDraft)

Loan (**L\_ID**、L\_Money、Count)

Pay (**P\_ID**、PayDate、PayMoney、B\_Name、L\_ID)

CCBAssociation (**C\_ID**、**B\_Name**、Co\_ID)

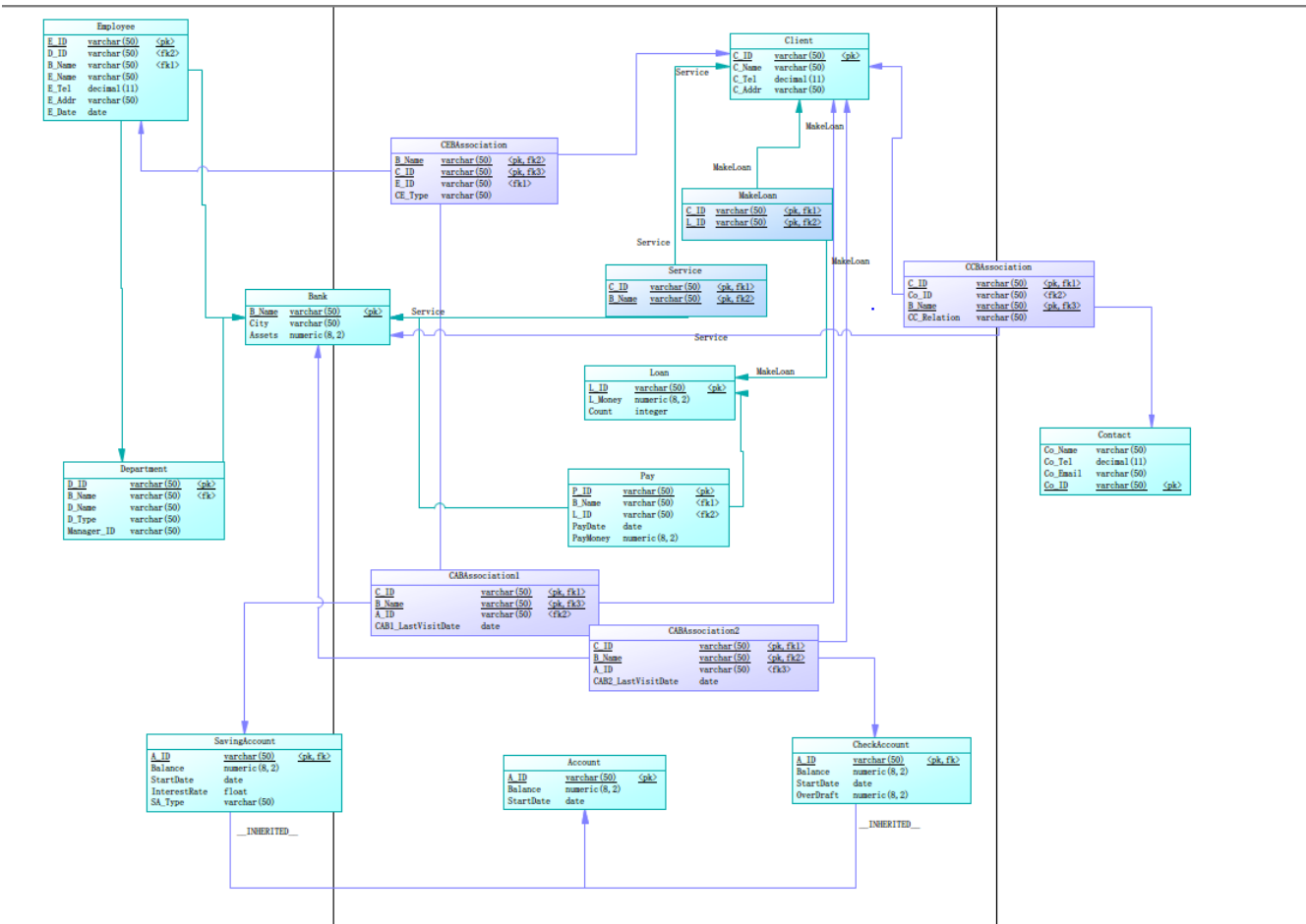
CEBAssociation (**B\_Name**、**C\_ID**、E\_ID)

CABAssociation1 (**B\_Name**、**C\_ID**、A\_ID)

CABAssociation2 (**B\_Name**、**C\_ID**、A\_ID)

### 3 MySQL数据库结构实现

#### 3.1 Power Designer的PDM图



#### 3.2 数据库表定义

##### 3.2.1 Bank表

列名	中文含义	类型	允许为空	是否主键	是否外键
B_Name	支行名	varchar(50)	否	是	否
City	支行所在城市	varchar(50)	否	否	否
Assets	支行资产	numeric(8,2)	否	否	否

##### 3.2.2 Client表

列名	中文含义	类型	允许为空	是否主键	是否外键
C_ID	客户身份证	varchar(50)	否	是	否
C_Name	客户名	varchar(50)	否	否	否
C_Tel	客户电话	decimal(11)	是	否	否
C_Address	客户住址	varchar(50)	是	否	否

### 3.2.3 Employee表

列名	中文含义	类型	允许为空	是否主键	是否外键
E_ID	员工身份证	varchar(50)	否	是	否
D_ID	部门号	varchar(50)	是	否	是(Department:D_ID)
B_Name	支行名	varchar(50)	否	否	是(Bank:B_Name)
E_Tel	员工电话	decimal(11)	是	否	否
E_Address	员工住址	varchar(50)	是	否	否
E_Begin_Work	开始工作时间	date	否	否	否

### 3.2.4 Department表

列名	中文含义	类型	允许为空	是否主键	是否外键
D_ID	部门号	varchar(50)	否	是	否
B_Name	支行名	varchar(50)	否	否	是(Bank:B_Name)
D_Name	部门名	varchar(50)	否	否	否
D_Type	部门类型	varchar(50)	否	否	否
Manager_ID	部门经理身份证	varchar(50)	否	否	否

### 3.2.5 Contact表

列名	中文含义	类型	允许为空	是否主键	是否外键
Co_ID	联系人身份证	varchar(50)	否	是	否
Co_Name	联系人姓名	varchar(50)	否	否	否
Co_Email	联系人邮箱	varchar(50)	是	否	否
Co_Tel	联系人电话	varchar(50)	否	否	否

### 3.2.6 Account表

列名	中文含义	类型	允许为空	是否主键	是否外键
A_ID	账户号	varchar(50)	否	是	否
Balance	账户余额	varchar(50)	否	否	否
StartDate	开户时间	date	否	否	否

### 3.2.7 SavingAccount表

列名	中文含义	类型	允许为空	是否主键	是否外键
A_ID	账户号	int	否	是	是(Account:A_ID)
Balance	账户余额	varchar(50)	否	否	否
StartDate	开户时间	date	否	否	否
S_Rate	利率	float	否	否	否
S_Type	储蓄类型	varchar(50)	否	否	否

3.2.8 CheckAccount表

列名	中文含义	类型	允许为空	是否主键	是否外键
A_ID	账户号	int	否	是	是(Account:A_ID)
Balance	账户余额	varchar(50)	否	否	否
StartDate	开户时间	date	否	否	否
C_OverDraft	透支额	numeric(8,2)	否	否	否

3.2.9 Loan表

列名	中文含义	类型	允许为空	是否主键	是否外键
L_ID	贷款号	varchar(50)	否	是	否
L_Money	贷款金额	numeric(8,2)	否	否	否
Count	取钱次数	integer	否	否	否

3.2.10 Pay表

列名	中文含义	类型	允许为空	是否主键	是否外键
P_ID	支付号	varchar(50)	否	是	否
B_Name	支行名	varchar(50)	否	否	是(Bank:B_Name)
L_ID	贷款号	varchar(50)	否	否	是(Loan:L_ID)
PayDate	支付日期	date	否	否	否
PayMoney	支付金额	numeric(8,2)	否	否	否

3.2.11 Service表

列名	中文含义	类型	允许为空	是否主键	是否外键
C_ID	客户身份证	varchar(50)	否	是	是(Client: C_ID)
B_Name	支行名	varchar(50)	否	是	是(Bank:B_Name)

3.2.12 MakeLoan表

列名	中文含义	类型	允许为空	是否主键	是否外键
C_ID	客户身份证	varchar(50)	否	是	是(Client: C_ID)
L_ID	借贷号	varchar(50)	否	是	是(Loan:L_ID)

3.2.13 CEBAssociation表

列名	中文含义	类型	允许为空	是否主键	是否外键
B_Name	支行名	varchar(50)	否	是	是(Bank:B_Name)
C_ID	客户身份证	varchar(50)	否	是	是(Client: C_ID)
E_ID	员工身份证	varchar(50)	否	否	是(Employee:E_ID)

3.2.14 CCBAAssociation表

列名	中文含义	类型	允许为空	是否主键	是否外键
B_Name	支行名	varchar(50)	否	是	是(Bank:B_Name)
C_ID	客户身份证	varchar(50)	否	是	是(Client: C_ID)
Co_ID	联系人身份证	varchar(50)	否	否	是(Contact:Co_ID)

### 3.2.15 CABAssociation1表

列名	中文含义	类型	允许为空	是否主键	是否外键
C_ID	客户身份证	varchar(50)	否	是	是(Client: C_ID)
B_Name	支行名	varchar(50)	否	是	是(Bank:B_Name)
A_ID	账户号	varchar(50)	否	否	是(SavingAccount:A_ID)

### 3.2.16 CABAssociation2表

列名	中文含义	类型	允许为空	是否主键	是否外键
C_ID	客户身份证	varchar(50)	否	是	是(Client: C_ID)
B_Name	支行名	varchar(50)	否	是	是(Bank:B_Name)
A_ID	账户号	varchar(50)	否	否	是(CheckAccount:A_ID)

## 3.3 总结与体会

通过此次实验实际体验到根据用户需求的描述来设计一个关系数据库模式,体会到在关系模式设计中的困难:实体和关系不易区分,总感觉有多种划分方案都可以,对用户需求的理解可能存在问题,另外在设计ER图时,在确定实体之间的关系是多对1还是多对多与实体选择的标识和业务需求有关,这点直接从最终需要的关系模式倒推感觉更容易,最后一点就是power design不是那么好用,使用图形拖拽的方式编程,虽然方便,但不太容易理解什么图形可以互相连接,连接后又代表什么含义,在使用过程中常常遇到找不到所需的图例,造成麻烦。