

lab2 银行业务管理系统 实验报告

- 实验环境

- 操作系统: Windows 10
- 建模工具: PowerDesigner 16.5

- 实验内容

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。

- 实验需求

对银行业务系统需求中的实体、实体属性以及实体之间的关联进行确认，以便画出正确的概念模型，进而导出物理模型和Oracle文件。

- 实验分析和设计过程

综合分析该银行业务管理需求，整理之后可以得到如下的实体表和实体联系表。

- 实体表

序号	名称	基本属性
1	支行	名字、城市、资产
2	部门	部门号、部门名称、部门类型、经理身份证号
3	员工	身份证号、姓名、住址、电话号码、开始工作日期
4	经理	（作为员工的超类）
5	账户	账户号、余额、开户日期、最近访问日期、开户支行
6	存储账户	利率、货币类型（作为账户的超类）
7	支票账户	透支额（作为账户的超类）
8	客户	身份证号、姓名、联系电话、家庭住址
9	客户联系人	姓名、手机号、邮箱、与客户的关系（弱实体）
10	贷款	贷款号金额、金额
11	贷款支付	支付金额、支付时间（弱实体）

o 实体联系表

序号	相关实体	联系类型	说明
1	支行：部门	1:n	一个支行有多个部门
2	部门：员工	1:n	一个部门有多名员工
3	员工：帐户	1:n	一个员工负责多个账户
4	经理：员工	1:n	一个经理管理多个员工
5	客户：账户	m:n	客户有多个账户，账户也可以被多人共有
6	客户：贷款	m:n	客户有多笔贷款，贷款也可以被多人共有
7	员工：贷款	1:n	一个员工负责多笔贷款
8	支行：贷款	1:n	一个支行发放多笔贷款

这样，实体和其拥有的属性以及实体之间的联系都确定，而且表明了几类特殊的实体，其中储蓄账户和支票账户为超类，贷款支付和联系人为弱实体，这几个实体在画cdm时要特殊考虑。

但是此时还剩下一个条件没有能够在上表中列出，在要求中提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，对于此处我的解决办法是在储蓄账户和支票账户中加入了一个客户身份证号的属性（可以唯一标识一个客户），同时将父类中的开户支行分别移到两种账户实体中去，然后将这两个属性联合设为Identifier，这样在储蓄账户和支票账户两个数据库表中就不会允许出现开户支行+客

户身份证号重复的记录，这样保证了每个用户在每个支行最多开设一个储蓄账户和一个支票账户，同时又不会违背一个客户可以开设多个账户的条件（在不同支行开设多个账户是允许的）。

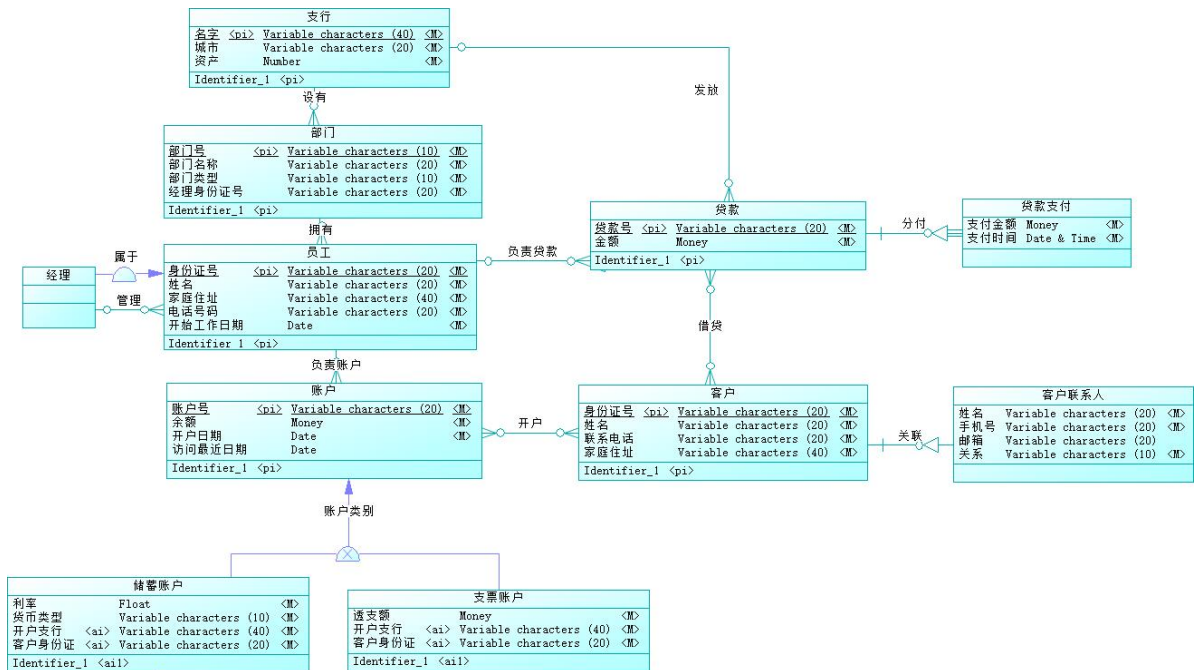
经过以上处理之后，设计出的数据库模型符合所有的需求，然后在powerdesigner中具体实现表示出来即可。

• 实验结果

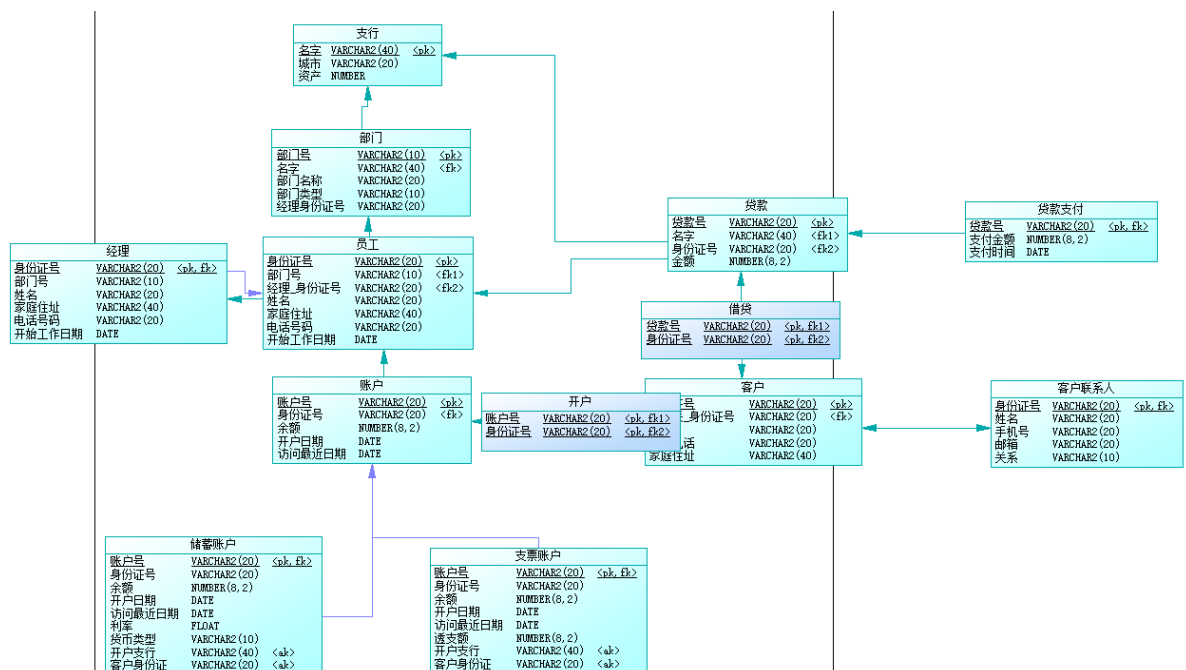
在powerdesig得到概念数据库模型cdm，生成Oracle物理数据库模型pdm和sql语句文件。

(清晰的cdm图和cdm文件、pdm文件以及sql源码文件已放在同目录下。)

◦ cdm



◦ pdm



- sql

部分截图如下：

```
/*=====*/
/* Table: "account" */
/*=====*/
create table "account"
(
    "account_id"          VARCHAR2(20)          not null,
    "staff_id"            VARCHAR2(20),
    "account_money"        NUMBER(8,2)          not null,
    "account_date"         DATE                  not null,
    "account_recent_date" DATE,
    constraint PK_ACCOUNT primary key ("account_id")
);

/*=====*/
/* Index: "account_in_charge_FK" */
/*=====*/
create index "account_in_charge_FK" on "account" (
    "staff_id" ASC
);

/*=====*/
/* Table: "borrow" */
/*=====*/
create table "borrow"
(
    "loan_id"             VARCHAR2(20)          not null,
    "client_id"           VARCHAR2(20)          not null,
    constraint PK_BORROW primary key ("loan_id", "client_id")
);
```

- 实验总结

本次实验分析了一个较为复杂的管理系统的数据库需求，然后从中整理出实体、属性以及各个实体之间的联系，通过实验，完整的实现了实体关系分析、数据库建模，并且使用powerdesigner将数据库建模进行实现，从而生成概念模型和物理模型。