银行业务管理系统数据库设计

PB20000180 刘良宇

1 概念模型设计

1.1 实体设计

1.1.1 支行：支行名字，支行城市，支行资产

1.1.2 部门：部门号，经理身份证，部门名称，部门类型

1.1.3 员工：员工身份证，员工姓名，员工电话，员工家庭地址，开始工作日期，是否部门经理

1.1.4 账户：账户号，账户余额，开户日期，最近访问日期，是否储蓄账户，利率，货币类型，透支额

1.1.5 客户：客户身份证号，客户姓名，客户电话，客户住址，联系人姓名，联系人手机号，联系人邮箱，联系人关系

1.1.6 贷款：贷款号，贷款金额

1.1.7 贷款发放：流水号，付款日期，付款金额

设计理由：

（1）客户联系人作为单独实体冗余，可以合并到客户实体。

（2）对于员工类型和账户类型，相当于存在不同的子类，但因为类型较少，可以枚举。因此这两处都采用布尔变量来表示类型，并将对应的属性设置为可选属性。

1.2 联系设计

支行和账户（开设）：一对多。一个多行管理多个账户。

支行和贷款（发放）：一对多。

拨付贷款：一对多，一笔贷款分成多次流水。

支行和部门（管理）：一对多。

部门和员工：一对多。

员工和客户：多对多，可能表示不同账户或借款的负责人。

账户和客户：多对多。一个账户可能为多个用户所有，一个用户也可以拥有多个账户。

贷款和用户：多对多。一笔贷款可能为多个用户所有，一个用户也可以有多笔贷款。

1.3 Power Designer 的 ER 图

基于前述分析，利用 Power Designer 设计了银行业务管理系统的数据库概念模型，结果如图 1 所示。

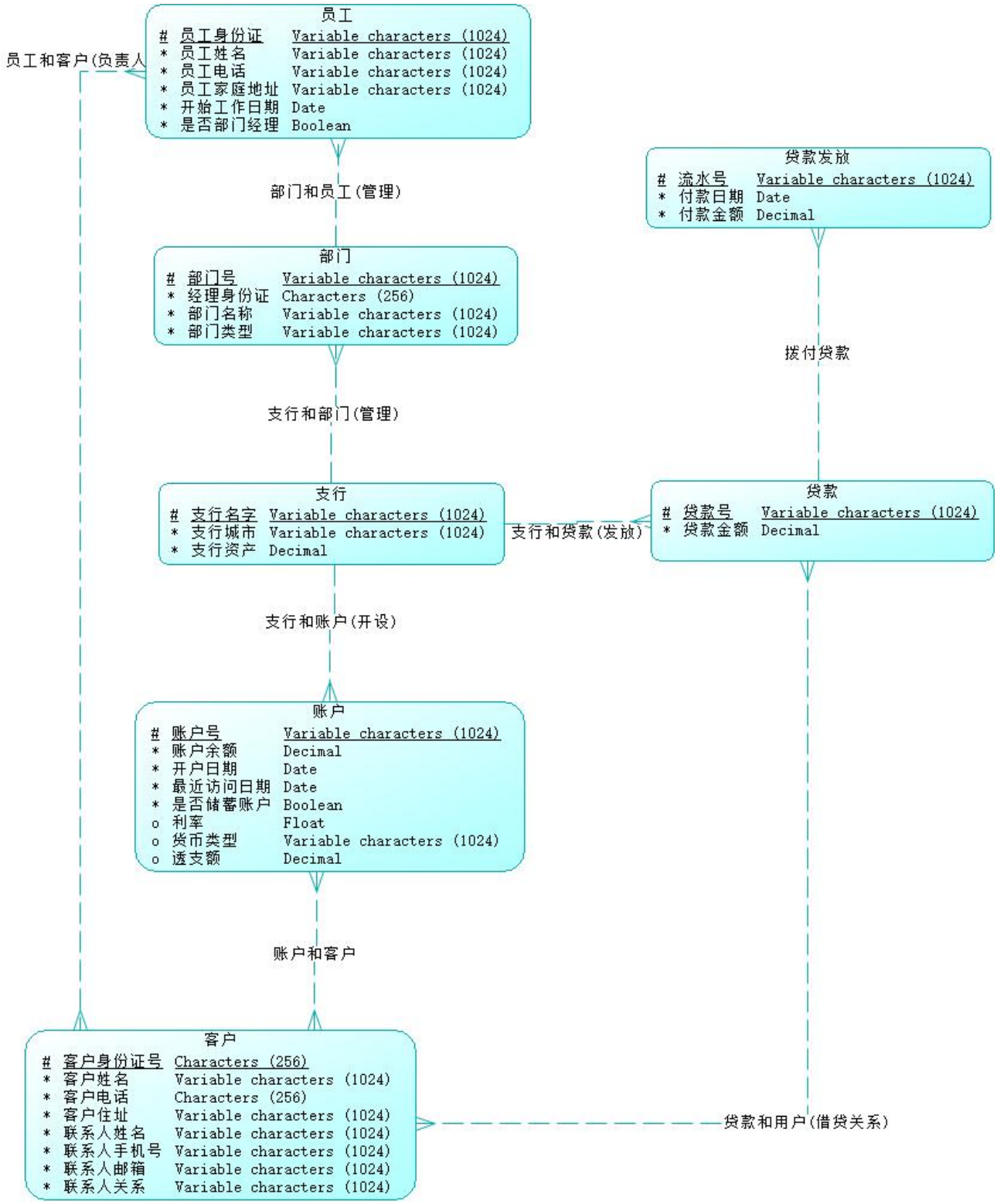

图 1. 银行业务管理系统的数据库概念模型

2 概念模型到逻辑模型的转换

2.1 实体转换

标识转为主码即可。

2.1.1 支行：支行名字，支行城市，支行资产

2.1.2 部门：部门号，经理身份证，部门名称，部门类型

2.1.3 员工：员工身份证，员工姓名，员工电话，员工家庭地址，开始工作日期，是否部门经理

2.1.4 账户：账户号，账户余额，开户日期，最近访问日期，是否储蓄账户，利率，货币类型，透支额

2.1.5 客户：客户身份证号，客户姓名，客户电话，客户住址，联系人姓名，联系人手机号，联系人邮箱，联系人关系

2.1.6 贷款：贷款号，贷款金额

2.1.7 贷款发放：流水号，付款日期，付款金额

2.2 联系转换

一对多关系分别作为外码，主码即可；

多对多关系，二者分别都是外码，且二者合并作为主码。

支行和账户（开设）：账户号主码，支行号外码，其他略

支行和贷款（发放）：贷款号主码，支行号外码，其他略

拨付贷款：流水号主码，贷款号外码，其他略

支行和部门（管理）：部门号主码，支行号外码，其他略

部门和员工：员工号主码，部门号外码，其他略

员工和客户：员工号和客户号是外码，共同作为主码

账户和客户：账户号和客户号是外码，共同作为主码

贷款和客户：贷款号和客户号是外码，共同作为主码

2.3 最终的关系模式

合并上面两个小节即可。

3 MySQL 数据库结构实现

3.1 Power Designer 的 PDM 设计

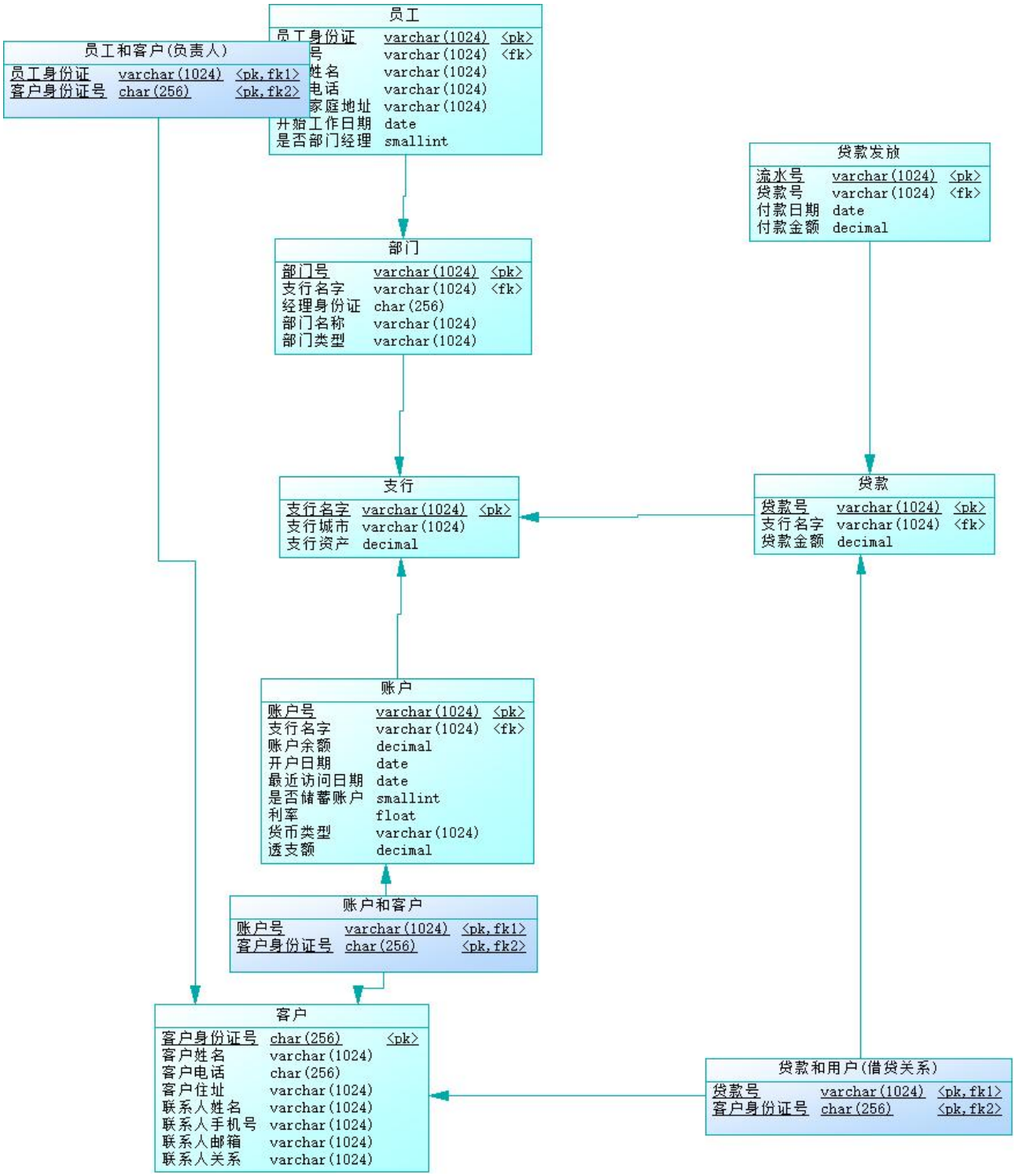

图 2 银行业务管理系统的 PDM 设计结果

3.2 数据库表定义

Power Designer 的 PDM 可以直接转换为 MySQL 中的基本表。下面给出了基于 PDM 构建的 MySQL 基本表设计结果，见如下图片。

```
/*==============================================================*/
/* Table: Bank                                                  */
/*==============================================================*/
Execute
create table Bank
(
   bank_name            varchar(1024)                  not null,
   bank_city            varchar(1024)                  not null,
   bank_money           decimal                        not null,
   constraint PK_BANK primary key (bank_name)
);
```

```
/*==============================================================*/
/* Table: account                                               */
/*==============================================================*/
Execute
create table account
(
   account_id           varchar(1024)                   not null,
   bank_name            varchar(1024)                   null,
   account_money        decimal                         not null,
   account_date         date                            not null,
   account_active_date  date                            not null,
   account_is_saving    smallint                        not null,
   account_rate         float                           null,
   account_type         varchar(1024)                   null,
   account_m            decimal                         null,
   constraint PK_ACCOUNT primary key (account_id)
);
```

```
/*==============================================================*/
/* Table: customer                                              */
/*==============================================================*/
```

Execute

```
create table customer
(
   customer_id          char(256)                      not null,
   customer_name        varchar(1024)                  not null,
   customer_phone       char(256)                      not null,
   customer_address     varchar(1024)                  not null,
   contact_name         varchar(1024)                  not null,
   contact_phone        varchar(1024)                  not null,
   contact_email        varchar(1024)                  not null,
   contact_relation     varchar(1024)                  not null,
   constraint PK_CUSTOMER primary key (customer_id)
);
```

```
/*==============================================================*/
/* Table: account_and_customer                                  */
/*==============================================================*/
```

Execute

```
create table account_and_customer
(
   account_id           varchar(1024)                   not null,
   customer_id          char(256)                       not null,
   constraint PK_ACCOUNT_AND_CUSTOMER primary key clustered (account
);
```

```
/*==============================================================*/
/* Table: depart                                                */
/*==============================================================*/
```

Execute

```
create table depart
(
   depart_id            varchar(1024)                  not null,
   bank_name            varchar(1024)                  null,
   manager_id           char(256)                      not null,
   depart_tname         varchar(1024)                  not null,
   depart_type          varchar(1024)                  not null,
   constraint PK_DEPART primary key (depart_id)
);
```

```
/*==============================================================*/
/* Table: loan                                                  */
/*==============================================================*/
```
Execute
```
create table loan
(
   loan_id              varchar(1024)                  not null,
   bank_name            varchar(1024)                  null,
   load_money           decimal                        not null,
   constraint PK_LOAN primary key (loan_id)
);
```

```
/*==============================================================*/
/* Table: loan_and_customer                                     */
/*==============================================================*/
```
Execute
```
create table loan_and_customer
(
   loan_id              varchar(1024)                  not null,
   customer_id          char(256)                      not null,
   constraint PK_LOAN_AND_CUSTOMER primary key clustered (loan_id, customer_id)
);
```

```
/*==============================================================*/
/* Table: loan_time                                             */
/*==============================================================*/
```
Execute
```
create table loan_time
(
   loan_time_id         varchar(1024)                  not null,
   loan_id              varchar(1024)                  null,
   loan_time_date       date                           not null,
   loan_time_money      decimal                        not null,
   constraint PK_LOAN_TIME primary key (loan_time_id)
);
```

```
/*==============================================================*/
/* Table: staff                                                 */
/*==============================================================*/
```
Execute
```
create table staff
(
   staff_id             varchar(1024)                  not null,
   depart_id            varchar(1024)                  null,
   staff_name           varchar(1024)                  not null,
   staff_phone          varchar(1024)                  not null,
   staff_address        varchar(1024)                  not null,
   staff_date           date                           not null,
   staff_is_manager     smallint                       not null,
   constraint PK_STAFF primary key (staff_id)
);
```

```
/*==============================================================*/
/* Table: staff_and_customer                                    */
/*==============================================================*/
Execute
create table staff_and_customer
(
   staff_id             varchar(1024)                  not null,
   customer_id          char(256)                      not null,
   constraint PK_STAFF_AND_CUSTOMER primary key clustered (staff_id, customer_id)
);
```

外键约束：

```
alter table account
   add constraint FK_ACCOUNT_BANK_AND__BANK foreign key (bank_name)
      references Bank (bank_name)
      on update restrict
      on delete restrict;

alter table account_and_customer
   add constraint FK_ACCOUNT__ACCOUNT_A_ACCOUNT foreign key (account_id)
      references account (account_id)
      on update restrict
      on delete restrict;

alter table account_and_customer
   add constraint FK_ACCOUNT__ACCOUNT_A_CUSTOMER foreign key (customer_id)
      references customer (customer_id)
      on update restrict
      on delete restrict;

alter table depart
   add constraint FK_DEPART_BANK_AND__BANK foreign key (bank_name)
      references Bank (bank_name)
      on update restrict
      on delete restrict;

alter table loan
   add constraint FK_LOAN_BANK_AND__BANK foreign key (bank_name)
      references Bank (bank_name)
      on update restrict
      on delete restrict;

alter table loan_and_customer
   add constraint FK_LOAN_AND_LOAN_AND__LOAN foreign key (loan_id)
      references loan (loan_id)
      on update restrict
      on delete restrict;
```

```
alter table loan_and_customer
   add constraint FK_LOAN_AND_LOAN_AND__CUSTOMER foreign key (customer_id)
      references customer (customer_id)
      on update restrict
      on delete restrict;

alter table loan_time
   add constraint FK_LOAN_TIM_LOAD_AND__LOAN foreign key (loan_id)
      references loan (loan_id)
      on update restrict
      on delete restrict;

alter table staff
   add constraint FK_STAFF_DEPART_AN_DEPART foreign key (depart_id)
      references depart (depart_id)
      on update restrict
      on delete restrict;

alter table staff_and_customer
   add constraint FK_STAFF_AN_STAFF_AND_STAFF foreign key (staff_id)
      references staff (staff_id)
      on update restrict
      on delete restrict;

alter table staff_and_customer
   add constraint FK_STAFF_AN_STAFF_AND_CUSTOMER foreign key (customer_id)
      references customer (customer_id)
      on update restrict
      on delete restrict;
```

4 总结与体会

本报告给出了利用 Power Designer 进行一个银行业务管理系统数据库的基本过程，包括概念模型设计、概念模型到逻辑模型的转换以及最终的 MySQL 数据库结构实现。

设计过程中的一些个人体会如下：

（1）对于复杂模型的设计，首先需要调查分析清楚需求，才便于后续设计。

（2）使用工具可以很方便的支持数据库设计工作。