

第一次 “数据库原理及应用” 课程作业

针对房屋租赁管理系统开发,设计实现该系统数据库,实现业务数据管理。

1) 分析房屋租赁管理系统业务的基本数据需求, 使用 Power Designer 建模工具, 建立房屋租赁管理系统概念数据模型。

经过分析,主要实体为租房者、房主、出租房和租房记录。房主在平台上传出租房信息,平台展示出租房信息,租房者浏览出租房信息并选择购买,租房者和出租房之间通过租房记录相关联。

提出租房者和房主的公共父类用户,可作系统概念数据模型图如图 1 所示。其中,房主拥有出租房,为一对多关系;租房者与租房记录为一对多关系;出租房与租房记录为一对多关系。

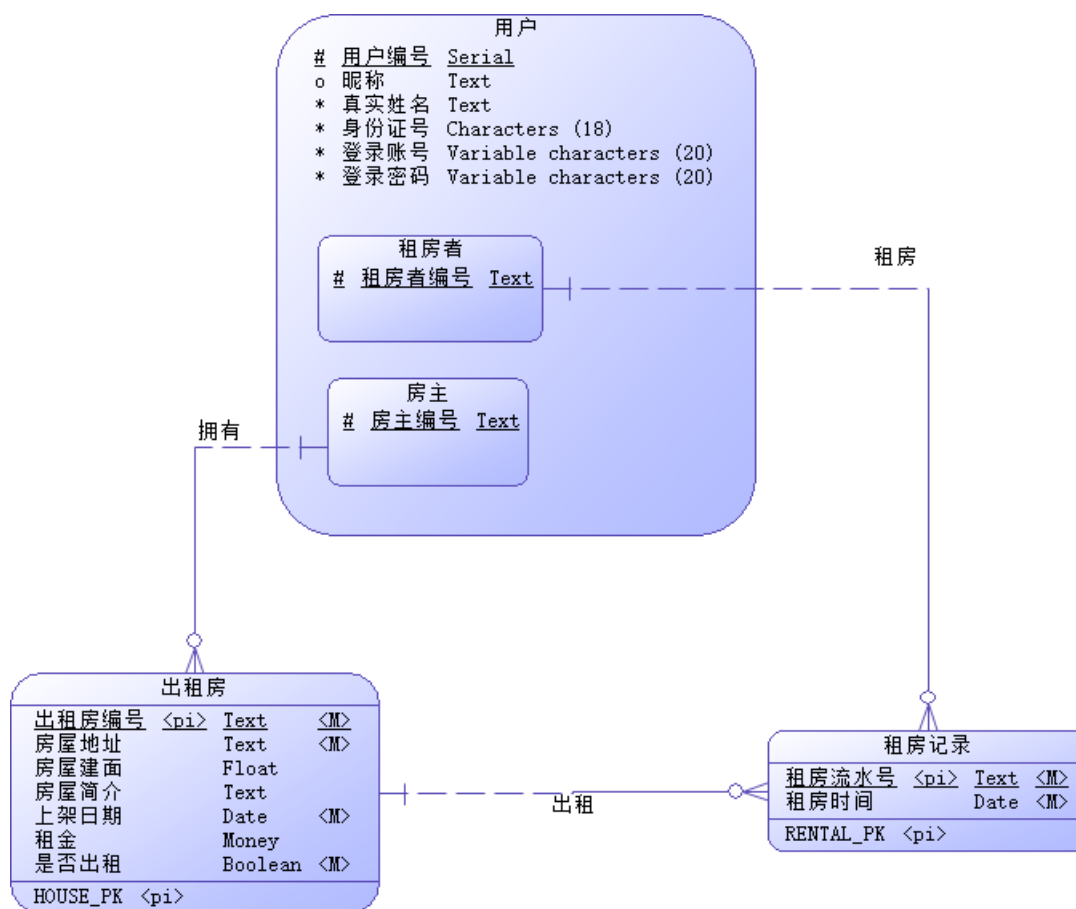


图 1 概念数据模型

2) 针对关系数据库设计, 在 Power Designer 建模工具中, 将房屋租赁管理系统概念数据模型转换为系统逻辑数据模型设计, 并进行规范化处理。

在图 1 的基础上, 作逻辑数据模型图如图 2 所示。对于用户实体, 用户编号为主键。房主和租房者实体继承用户实体, 分别设房主编号和租房者编号为另一个主键。出租房实体中, 出租房编号为主键, 房主编号 (或用户编号) 为外键。租房记录实体中, 租房流水号为主键, 出租房编号为第一外键, 租房者编号 (或用户编号) 为第二外键。

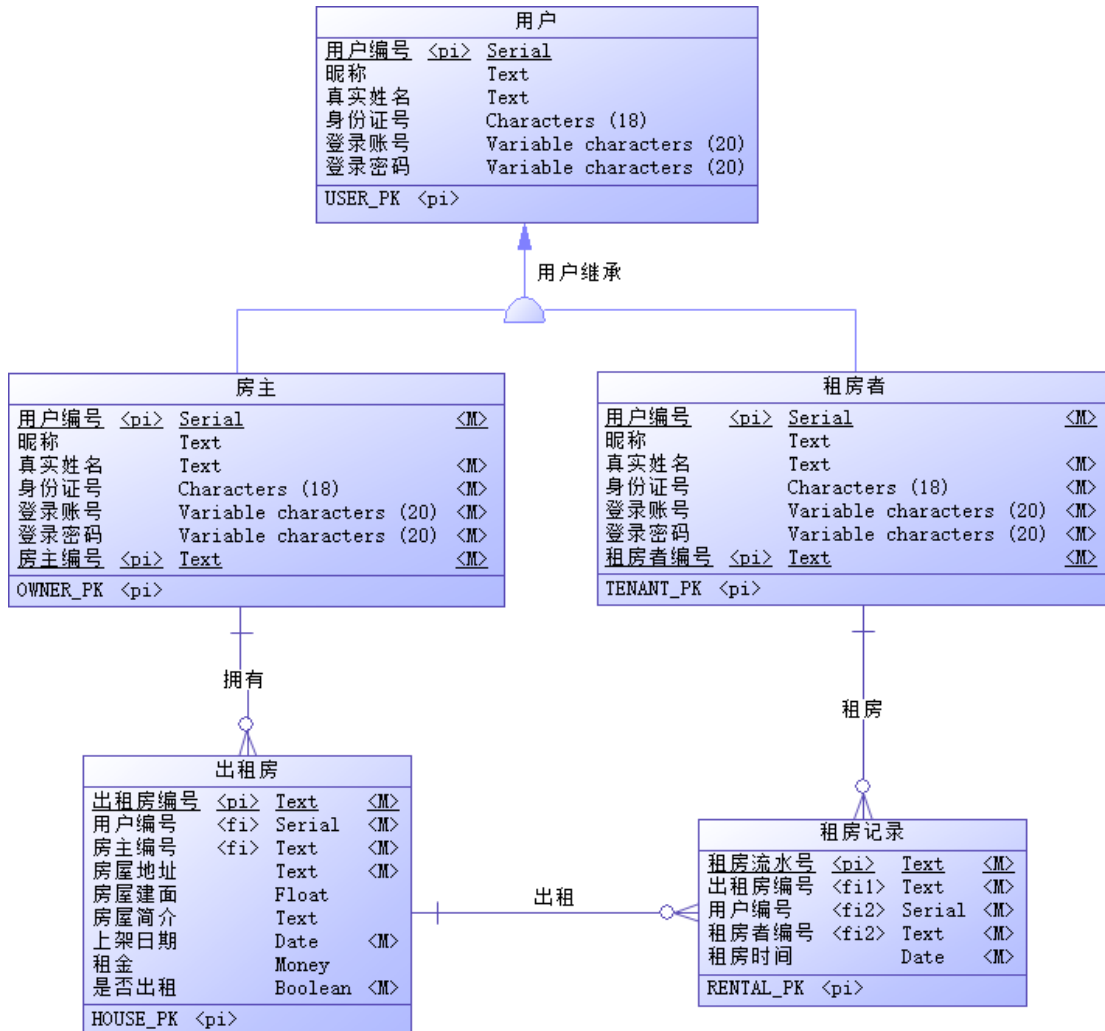


图 2 逻辑数据模型

3) 针对 PostgreSQL 数据库实现, 在 Power Designer 建模工具中, 将房屋租赁管理系统逻辑数据模型转换为系统物理数据模型设计。

进一步将图 2 转换为物理数据模型图如图 3 所示。房主和租房者实体中的用户编号既为主键, 也是外键。逻辑数据模型中的 Characters()、Variable characters()和 Boolean 均修改为 Postgresql 数据库中的术语, 分别为 CHAR()、VARCHAR()和 BOOL。

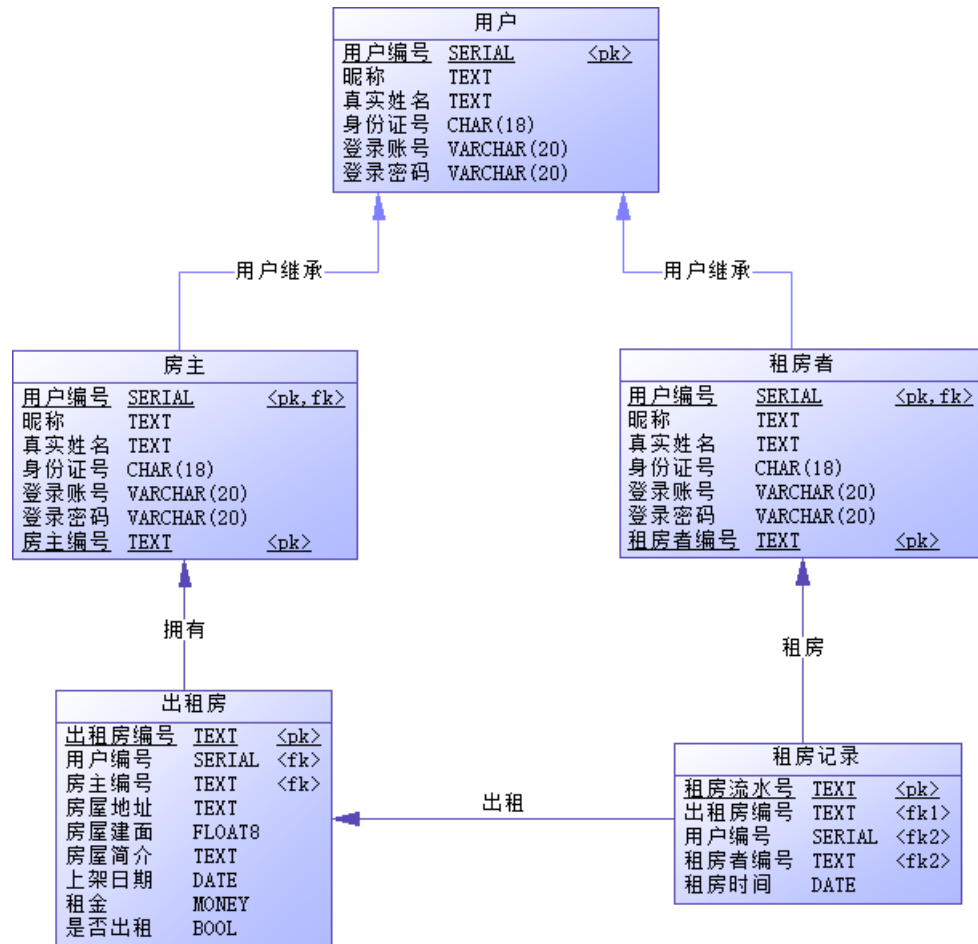


图 3 物理数据模型

4) 在 Power Designer 建模工具中, 将房屋租赁管理系统物理数据模型转换为 SQL 脚本。
转换后的 SQL 脚本如代码 1 所示。

代码 1 房屋租凭管理系统 SQL 脚本

```

/*=====*/
/* DBMS name:      PostgreSQL 9.x                      */
/* Created on:      2020/4/26 16:01:35                  */
/*=====*/

drop index 拥有_FK;

drop index 出租房_PK;

drop table 出租房;

drop index 用户继承_FK;

drop index 房主_PK;

drop table 房主;

drop index 用户_PK;

drop table 用户;

drop index 用户继承 2_FK;

drop index 租房者_PK;

drop table 租房者;

drop index 租房_FK;

drop index 出租_FK;

drop index 租房记录_PK;

drop table 租房记录;

/*=====*/
/* Table: 出租房                      */
/*=====*/
create table 出租房 (
    出租房编号          TEXT          not null,

```

```

        用户编号          SERIAL          not null,
        房主编号          TEXT            not null,
        房屋地址          TEXT            not null,
        房屋建面          FLOAT8          null,
        房屋简介          TEXT            null,
        上架日期          DATE            not null,
        租金              MONEY           null,
        是否出租          BOOL            not null,
        constraint PK_出租房 primary key (出租房编号)
    );

/*=====*/
/* Index: 出租房_PK                                     */
/*=====*/
create unique index 出租房_PK on 出租房 (
    出租房编号
);

/*=====*/
/* Index: 拥有_FK                                     */
/*=====*/
create index 拥有_FK on 出租房 (
    用户编号,
    房主编号
);

/*=====*/
/* Table: 房主                                     */
/*=====*/
create table 房主 (
    用户编号          SERIAL          not null,
    昵称              TEXT            null,
    真实姓名          TEXT            not null,
    身份证号          CHAR(18)        not null,
    登录账号          VARCHAR(20)     not null,
    登录密码          VARCHAR(20)     not null,
    房主编号          TEXT            not null,
    constraint PK_房主 primary key (用户编号, 房主编号)
);

/*=====*/
/* Index: 房主_PK                                     */
/*=====*/
create unique index 房主_PK on 房主 (

```

```

用户编号,
房主编号
);

```

```

/*=====*/
/* Index: 用户继承_FK                                     */
/*=====*/

```

```

create index 用户继承_FK on 房主 (
用户编号
);

```

```

/*=====*/
/* Table: 用户                                           */
/*=====*/

```

```

create table 用户 (
    用户编号          SERIAL          not null,
    昵称              TEXT            null,
    真实姓名          TEXT            not null,
    身份证号          CHAR(18)        not null,
    登录账号          VARCHAR(20)     not null,
    登录密码          VARCHAR(20)     not null,
    constraint PK_用户 primary key (用户编号)
);

```

```

/*=====*/
/* Index: 用户_PK                                       */
/*=====*/

```

```

create unique index 用户_PK on 用户 (
用户编号
);

```

```

/*=====*/
/* Table: 租房者                                         */
/*=====*/

```

```

create table 租房者 (
    用户编号          SERIAL          not null,
    昵称              TEXT            null,
    真实姓名          TEXT            not null,
    身份证号          CHAR(18)        not null,
    登录账号          VARCHAR(20)     not null,
    登录密码          VARCHAR(20)     not null,
    租房者编号        TEXT            not null,
    constraint PK_租房者 primary key (用户编号, 租房者编号)
);

```

```

/*=====*/
/* Index: 租房者_PK                                     */
/*=====*/
create unique index 租房者_PK on 租房者 (
  用户编号,
  租房者编号
);

/*=====*/
/* Index: 用户继承 2_FK                                     */
/*=====*/
create index 用户继承 2_FK on 租房者 (
  用户编号
);

/*=====*/
/* Table: 租房记录                                     */
/*=====*/
create table 租房记录 (
  租房流水号          TEXT          not null,
  出租房编号          TEXT          not null,
  用户编号            SERIAL        not null,
  租房者编号          TEXT          not null,
  租房时间            DATE          not null,
  constraint PK_租房记录 primary key (租房流水号)
);

/*=====*/
/* Index: 租房记录_PK                                     */
/*=====*/
create unique index 租房记录_PK on 租房记录 (
  租房流水号
);

/*=====*/
/* Index: 出租_FK                                     */
/*=====*/
create index 出租_FK on 租房记录 (
  出租房编号
);

/*=====*/
/* Index: 租房_FK                                     */

```

```
/*=====*/  
create index 租房_FK on 租房记录 (  
    用户编号,  
    租房者编号  
);  
  
alter table 出租房  
    add constraint FK_出租房_拥有_房主 foreign key (用户编号, 房主编号)  
        references 房主 (用户编号, 房主编号)  
        on delete restrict on update restrict;  
  
alter table 房主  
    add constraint FK_房主_用户继承_用户 foreign key (用户编号)  
        references 用户 (用户编号)  
        on delete restrict on update restrict;  
  
alter table 租房者  
    add constraint FK_租房者_用户继承_2_用户 foreign key (用户编号)  
        references 用户 (用户编号)  
        on delete restrict on update restrict;  
  
alter table 租房记录  
    add constraint FK_租房记录_出租_出租房 foreign key (出租房编号)  
        references 出租房 (出租房编号)  
        on delete restrict on update restrict;  
  
alter table 租房记录  
    add constraint FK_租房记录_租房_租房者 foreign key (用户编号, 租房者编号)  
        references 租房者 (用户编号, 租房者编号)  
        on delete restrict on update restrict;
```


5) 在 PostgreSQL 数据库服务器中, 执行该 SQL 脚本, 实现房屋租赁管理系统数据库对象创建实现。

在 PostgreSQL 数据库服务器中执行上述脚本, 得到结果如图 4 至 8 所示。



图 4 租房房表

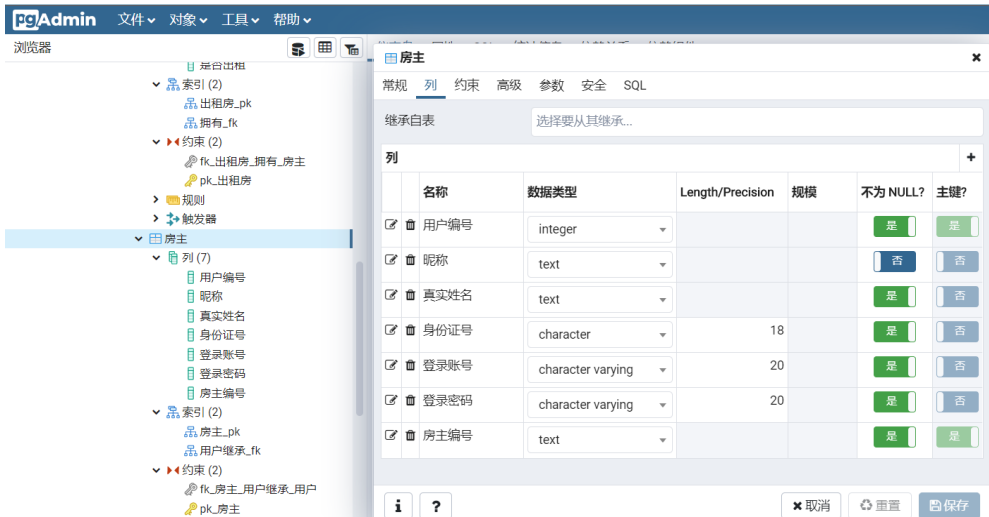


图 5 房主表

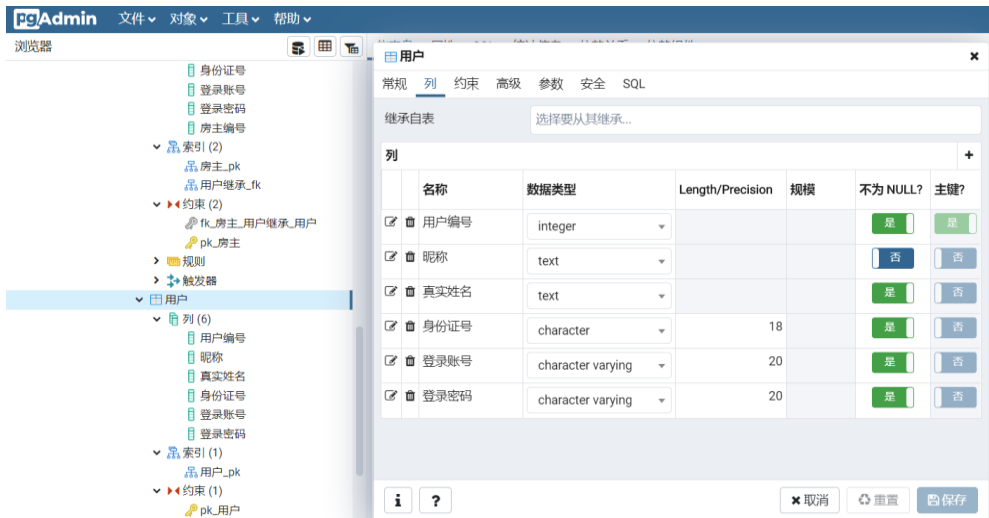


图 6 用户表

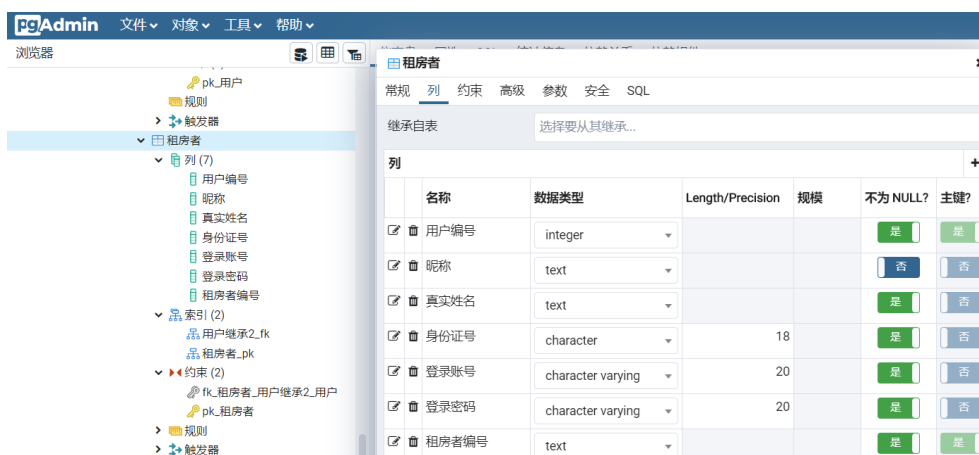


图 7 租房者表

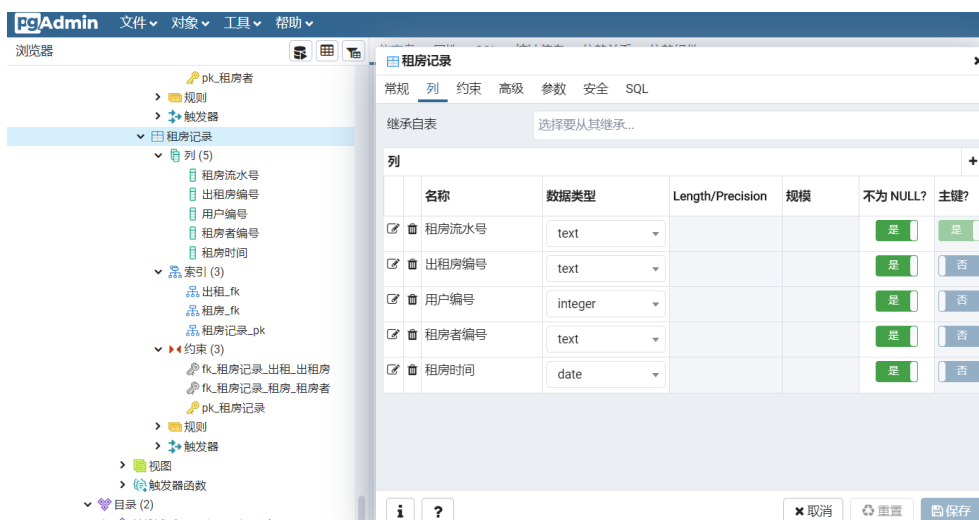


图 8 租房记录表

作业要求：在作业文档分别给出房屋租赁管理系统的概念数据模型、逻辑数据模型、物理数据模型模型设计图，并给出数据库设计在 PostgreSQL 数据库中实现。给出各个模型设计步骤、设计说明、执行结果界面，并对结果进行说明。

作业文件格式：作业 1_学号_姓名.doc

作业成绩评价标准：

正确完成情况	优	优	优	良	良	良	良	良	中	中
作业过程情况	优	优	良	优	良	良	良	中	中	中
文档规范性	优	良	良	优	优	良	中	中	中	差
作业评分	100-98	97-95	94-92	91-89	88-85	84-82	81-79	78-76	75-73	72-70

作业提交：

<http://www.wlxt.uestc.edu.cn/wlxt/course.aspx?courseid=0567>