



哈爾濱工業大學 (深圳)  
HARBIN INSTITUTE OF TECHNOLOGY

# 实验报告

开课学期: 2021 春季  
课程名称: 数据库系统  
实验名称: 二手房销售系统设计与实现  
实验性质: 设计型  
实验学时: 2 地点: T2608  
学生班级: 1801101  
学生学号: 180110115  
学生姓名: 方澳阳  
评阅教师:  
报告成绩:

实验与创新实践教育中心制

2021 年 1 月

# 1 实验环境

Ubuntu 20.04

Mysql Ver 8.0.23-0ubuntu0.20.04.1 for Linux on x86\_64

Goland 2021.3

使用 web 框架: beego

# 2 实验过程

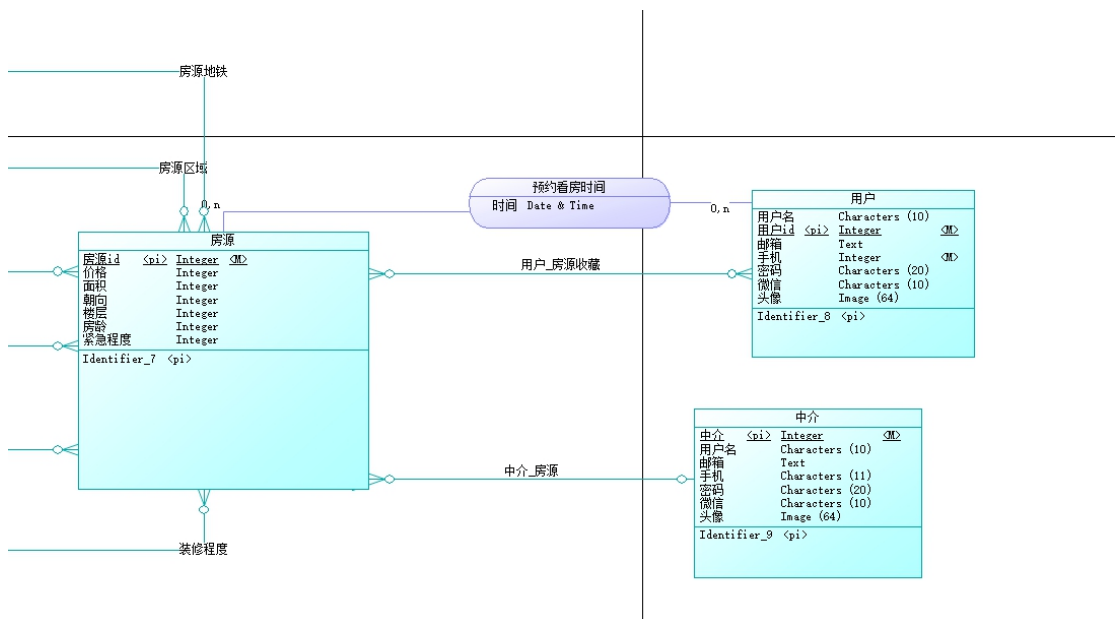
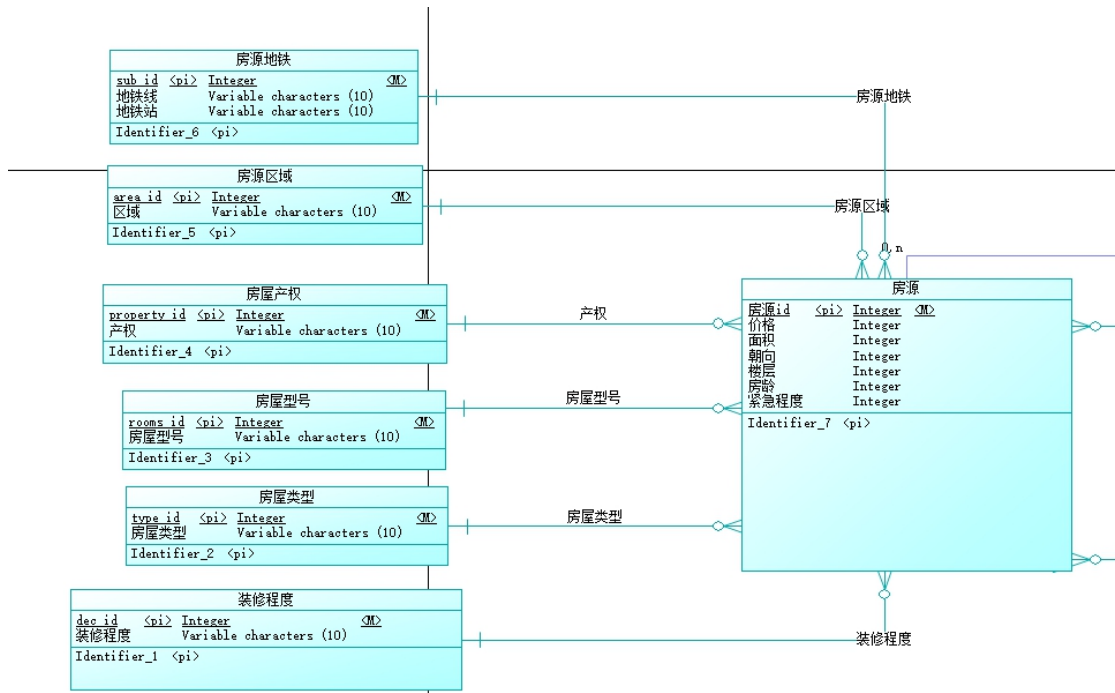
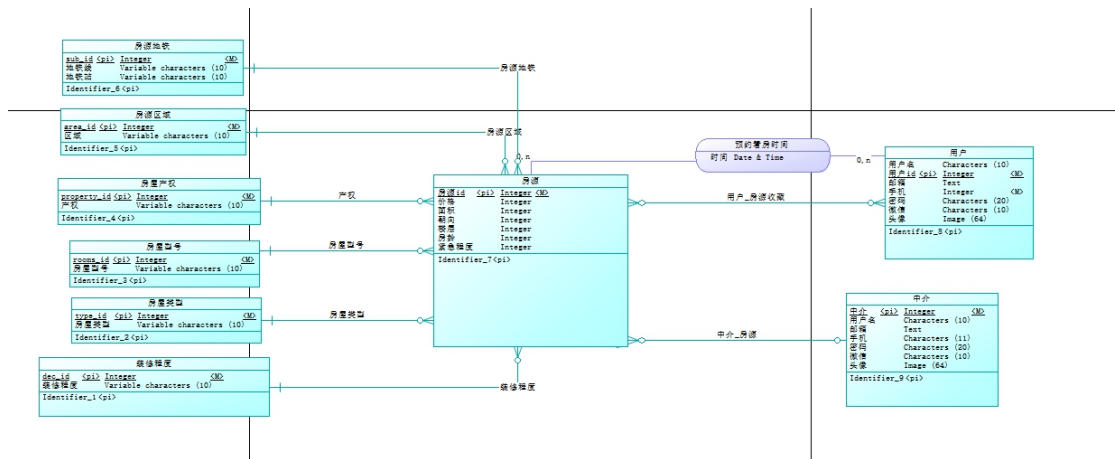
## 2.1 系统功能

实现功能	备注
用户登录	区分了不同的用户（买方、卖方），有不同的权限
用户修改信息	用户注册后可以修改个人信息
用户退出	
卖方发布房屋信息	买方无法发布，需要注册卖方账号
卖方删除房屋信息	
卖方查看自己拥有的房屋信息	
买方、卖方查看所有的房屋信息	
买方收藏房屋信息	
买方预约房屋信息	
买方取消预约和收藏	

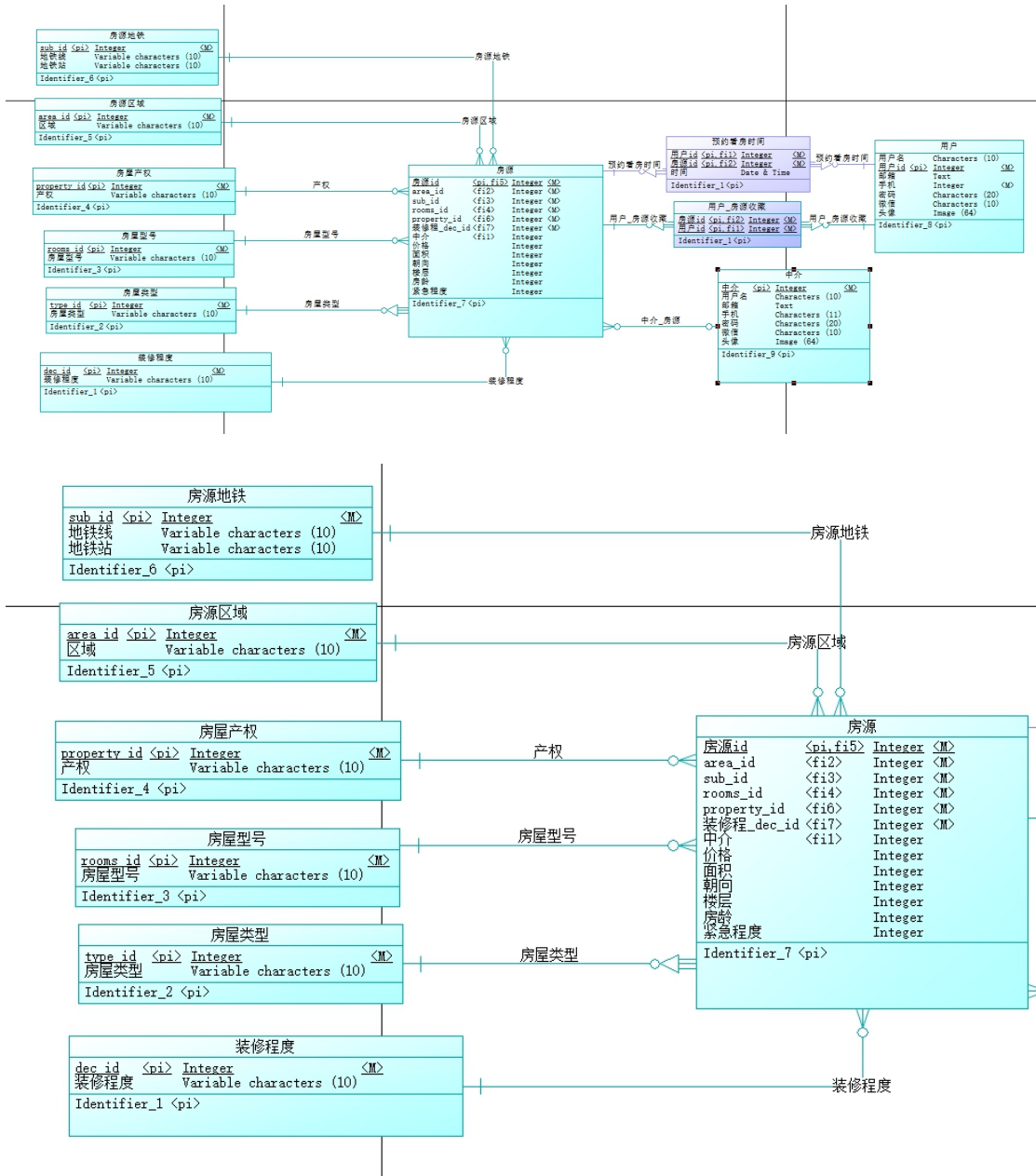
## 2.2 数据库设计

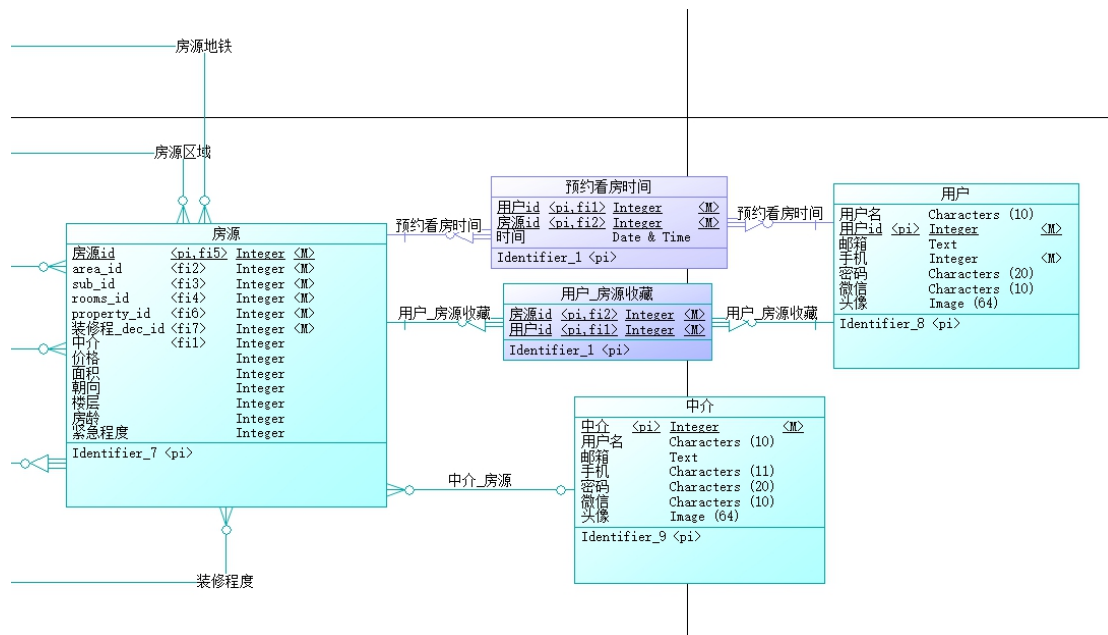
### 2.1.1 ER 图

要求：截图务必清晰，如果图太大可截图一个总图，然后截几个部分图展示。如果看不清截图会影响成绩。

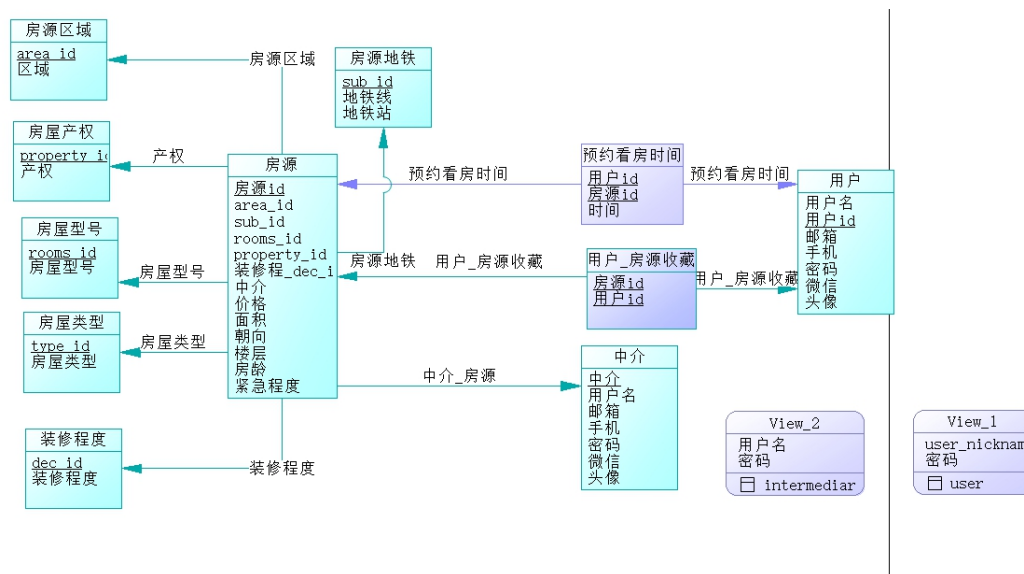


2.1.2 LDM 图





## 2.1.3 PDM 图



## 2.1.4 数据库表结构

### 1、 表结构

Id 被默认设为主键。Direction\Floor\Age\Emergency 体现了空值约束。剩余的体现了外键约束。

fk: foreign key, 外键, 一对多关系

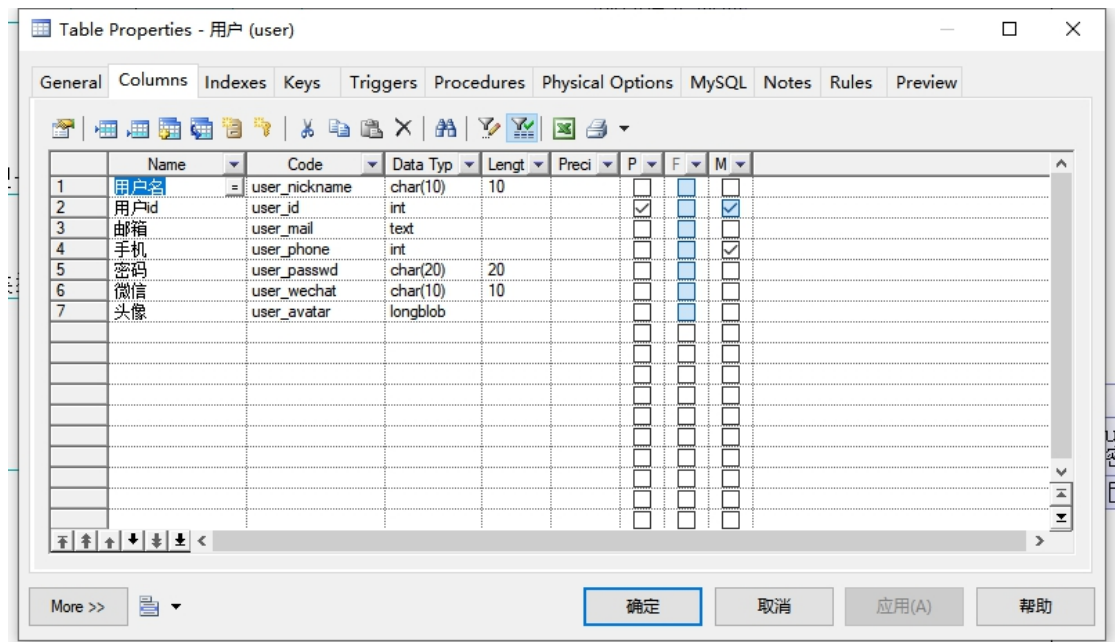
m2m: many to many, model to model, 多对多关系

```
71 type House struct {
72     Id int
73     Space int `orm:"null"`
74     Price int
75     Direction int `orm:"null"`
76     Floor int `orm:"null"`
77     Age int `orm:"null"`
78     Emergency int `orm:"null"`
79     Subway *Subway `orm:"rel(fk)"`
80     Area *Area `orm:"rel(fk)"`
81     Property *Property `orm:"rel(fk)"`
82     Rooms *Rooms `orm:"rel(fk)"`
83     HouseType *HouseType `orm:"rel(fk)"`
84     Decoration *Decoration `orm:"rel(fk)"`
85     Inter *Inter `orm:"rel(fk)"`
86     Favourites []*User `orm:"rel(m2m)"`
87     Appointments []*Appointment `orm:"reverse(many)"`
88 }
```

	Name	Code	Data Type	Length	Preci	P	F	M
1	房源id	house_id	int			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	area_id	area_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	sub_id	sub_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	rooms_id	rooms_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	property_id	property_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	装修程_dec_id	hou_house_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	中介	zj_id	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	价格	house_price	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	面积	house_space	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	朝向	house_dir	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	楼层	house_floor	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	房龄	house_age	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	紧急程度	house_emr	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

这里同理

```
13 type User struct {
14     Id      int
15     Name    string    `orm:"index"`
16     Pwd     string    `orm:"size(20)"`
17     PhoneNum string    `orm:"index;unique;null"`
18     Mail    string    `orm:"index;unique;null"`
19     Wechat  string    `orm:"unique;null"`
20     Avatar  string    `orm:"size(256);null"`
21     Appointments []*Appointment `orm:"reverse(many)"`
22     Favourites []*House    `orm:"reverse(many)"`
23 }
```



## 2、索引

我将用户登录的用户名、邮箱、手机号建立了非主索引。因为用户登录时通常会通过这些方式登录，在用户数量非常多的时候有利于加速。

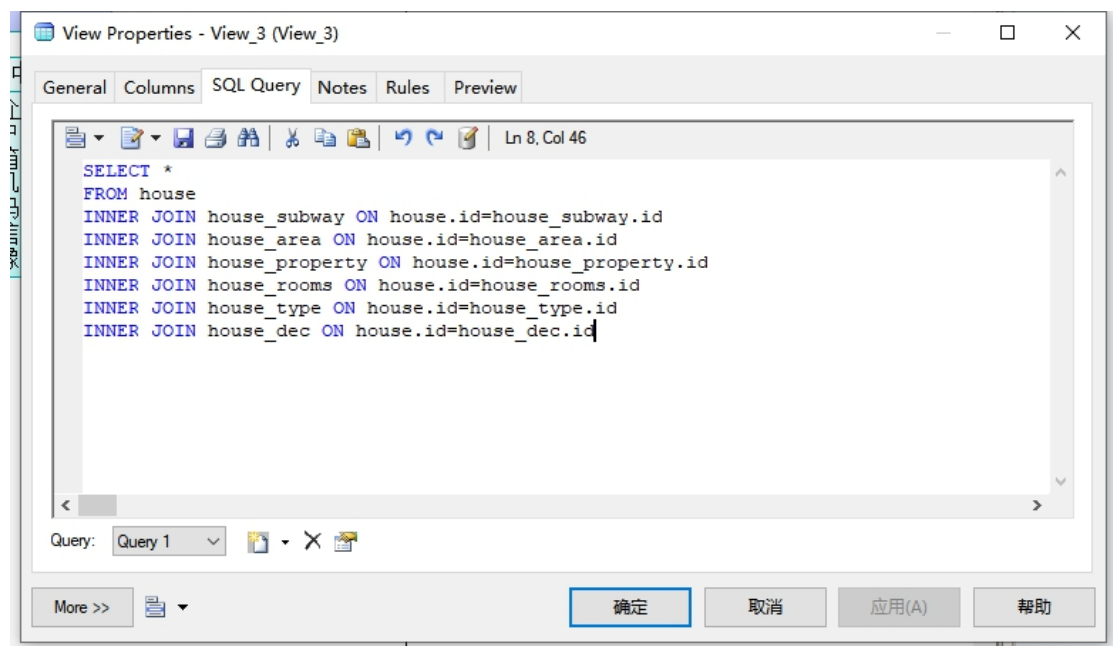
中介登录同理。

```
// 多字段索引
func (u *User) TableIndex() [][]string {
    return [][]string{
        []string{"Id", "Name"},
    }
}

// 多字段唯一键
func (u *User) TableUnique() [][]string {
    return [][]string{
        []string{"Name", "Email"},
    }
}
```

### 3、 视图

讲解你建的视图，说明为什么要建这个视图。



在用户查看表的详情时，需要连接很多表。因此建立这个视图



## 4、 事务（选做）

在中介发布二手房信息时，他输入的数据将被写到不同的表中，如果此时机器发生宕机，则容易造成不一致。所以设计一个事务来保证数据库的一致性。

伪码如下：

```
orm.Begin()
err:=插入 house
if err != null {
    orm.Rollback()
}
err:=插入 subway
if err != null {
    orm.Rollback()
}
err:=插入 area
if err != null {
    orm.Rollback()
}
...
orm.Commit()
```

## 2.1.5 分析

### 1. 理解需求，确定实体

- (1) 用户需要区分买方和卖方
- (2) 买方和买方有不同的权限
- (3) 房子有不同的信息

房子有价格、地区、金额、面积、朝向、房龄、房间数、楼层等等一系列属性。这些属性是一对多的。比如同一地区有很多个房子，同一楼层也有很多的房子，但房子只有一个楼层，只能在一个地区。

但针对一些数字类型的数值来说，将他们分出一个表来的代价比较大。因为查询他们的时候，通常是通过区间来查询的，比如查询价格在 100 万到 200 万之间的二手房。

而对于字符串类型的数值来说，比如地区、装修的程度、房间的规格（两室一厅），是比较适合分表的。因为如果这些属性全都在一个表里的话，首先可能会导致查询缓慢，其次还有非常多的数据冗余。

对于用户来说，他与房子的关系是多对多的。用户可以收藏、预约非常多的房间，而同一个房间也可以被很多用户预约和收藏。房子对于中介来说也是一对多的，一个中介可能很多房子，但房子一定只属于一个中介。因此可以确定了 ER 图的画法。

### 2. 画 ER 图

### 3. 生成 Idm、pdm

### 3 收获和反思

在较短的时间内学习了 web 框架的使用，了解了 web 应用的整套流程。但也因为是初学框架，导致并不熟练，也走了很多弯路。

对于数据库的设计思路有了更深的了解，但仍然是初级阶段。应该在以后更多的实践中来体会为什么分库分表、为什么设计视图、为什么设计索引，怎么设计事务来保证系统可用性的前提下，提高并发性。