

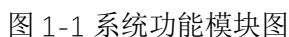
# 四川轻化工大学实验报告

院：计算机科学与工程学院 课程名称：数据库系统原理 日期：2022.5.11

姓名	李万余, 胡鹏, 罗荣	组号	5	学号	20XX	实验室	329
专业	软件工程	班号		班号	2020 级 4 班	老师签名	
实验名称	实验一、数据库系统分析与设计					成绩评定	
所用仪器材料	计算机, Word 2019, Visio 2016, MySQL 8.0.28, Navicat for MySQL 15						
实验目的或要求	<p>项目名称：物业管理系统</p> <p><b>一、实验目的</b></p> <ol style="list-style-type: none"> <li>1. 掌握数据库设计的基本技术。</li> <li>2. 熟悉数据库设计的每个步骤中的任务和实施方法,并加深对数据库系统概念和特点的理解。</li> <li>3. 熟练掌握使用 SQL 语言创建数据库、表、索引和修改表结构。</li> <li>4. 熟练掌握使用 SQL 语言向数据库输入数据、修改数据和删除数据的操作。</li> <li>5. 掌握通过 MySQL 命令行和 Navicat 观察创建结果,并进行分析。掌握 Java 面向对象程序设计的基本思想。</li> </ol> <p><b>二、实验内容及要求</b></p> <ol style="list-style-type: none"> <li>1. 要求学生根据周围的实际情况,自选一个小型的数据库应用项目并深入到应用项目的现实世界中,进行系统分析和数据库设计。例如选择学籍管理系统、图书管理系统、材料管理系统或仓库管理系统等。</li> <li>2. 用关系数据库理论对自己设计的数据模型进行评价,指出合理和不足之处,提出改进的方案。</li> <li>3. 通过 SQL 语句,创建数据库和查看数据库属性。创建表、确定表的主码和约束条件,为某些字段建立索引。查看和修改表结构。</li> <li>4. 通过 SQL 语句,在数据库的各个表中各输入 8~30 条记录。要求记录不仅满足数据约束要求,还要有表间关联的记录。</li> <li>5. 通过 MySQL 命令行完成上述操作,并用 Navicat 观察结果。</li> </ol> <p><b>1. 具体内容及要求:</b></p> <p>需求分析叙述合理、全面;在 E-R 图中注意实体间联系类型的表示;数据库设计时应避免数据冗余;数据库的表中,属性应是不可再分的最小项;逻辑设计应按照转化原则及完整性约束进行规范设计,各表的联系应通过外码实现。根据数据库逻辑设计结果,在数据库中进行每一个表的定义,并添加数据入库,编写每个表的 SQL 定义语句及数据添加语句,注意表格建立的顺序。</p>						

实  
验  
原  
理

本系统主要由后台发布管理和前台浏览评论两部分组成，其中后台管理包含用户管理、栏目管理、新闻管理、留言管理等相关模块，前台功能包含首页展示、新闻列表、新闻展示、新闻评论、用户注册及登录等相关功能。本系统的功能模块如图 2-1 所示。



### 2.1. 数据实体及相关的联系分析

1. 用户信息：用户 ID，密码，姓名，手机号，邮箱，激活状态；
2. 投诉/建议：投诉/建议 ID，投诉/建议时间，投诉/建议内容，被投诉/建议的对象；
3. 消息：消息 ID，发布时间，消息内容，消息有效期，消息类型；
4. 费用：费用 ID，开始时间，缴费时间，金额；
5. 费用类型：费用类型，费用类型 ID；
6. 维修任务：维修 ID，维修状态，维修内容；
7. 停车位：停车位 ID，停车位位置，购买时间，停车位状态；
8. 管理员信息：管理员 ID，密码，姓名，手机号，邮箱，激活状态；
9. 房屋：房屋 ID，房屋位置，房屋状态；

1. 一个用户可以发送多条投诉/建议，一条投诉/建议只能由一个用户发出，即 1:n 关系；
2. 一个管理员可以响应多条投诉/建议，一条投诉/建议只能被一个管理员响

应，即 1: n 关系；

3. 一个管理员可以发布多条消息，一条消息只能由一个管理员发布，即 1: n 关系；
4. 一个房屋可以产生多项费用，一项费用只能由一个房屋用户产生，即 1: n 关系；
5. 一个用户可以缴纳多项费用，一项费用只能由一个用户缴纳，即 1: n 关系；
6. 一个费用类型有多条费用，一条费用只属于一个费用类型，即 1: n 关系；
7. 一个用户可以发布多条维修任务，一个维修任务只能被一个用户发布，即 1: n 关系；
8. 一个管理员可以处理多条维修任务，一个维修任务可以被管理员多次处理，即 n: m 关系；
9. 一间房屋可以购买多个停车位，一个停车位只能被一个用户拥有，即 1: n 关系；
10. 一个用户可以拥有多个房屋，一个房屋可以属于多个用户，即 n: m 关系
11. 一个管理员可以更新多个房屋信息，一个房屋信息可以被一个管理员多次更新，即 n: m 关系；
12. 一个上级管理员可以有多个子管理员，一个下级管理员只能有一个上级管理员，即 1: n 关系；

## 2.2. E-R 模型设计

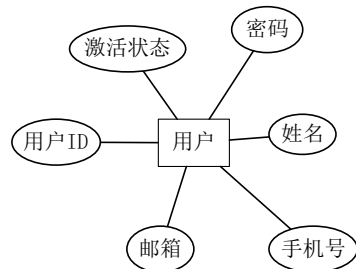


图 2-1 用户实体图

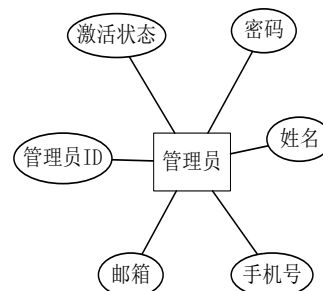


图 2-2 管理员实体图

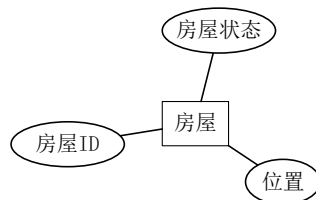


图 2-3 房屋实体图

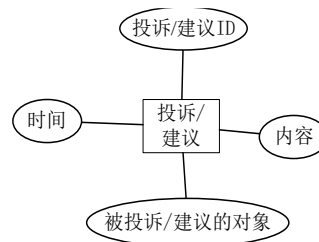


图 2-4 投诉/建议实体图

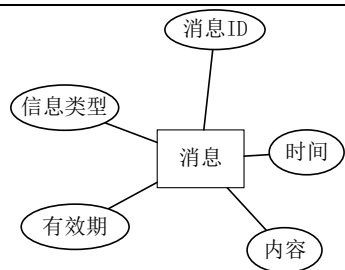


图 2-5 消息实体图

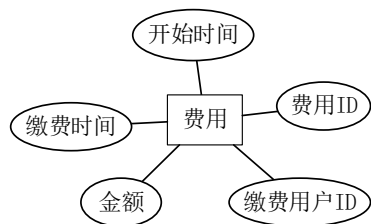


图 2-6 费用实体图

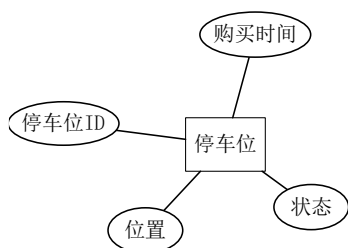


图 2-7 停车位实体图

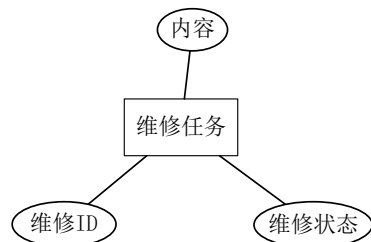


图 2-8 维修任务实体图

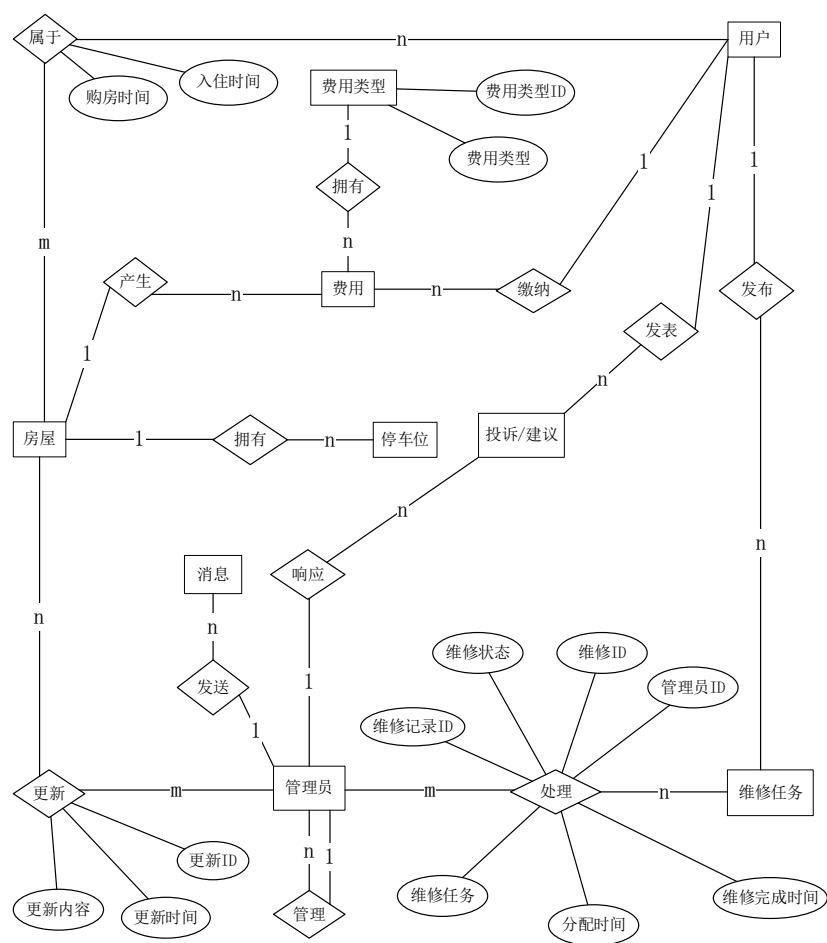


图 2-8 系统总体 E-R 图

### 3. 逻辑结构设计

#### 3.1. 关系模式建立

概念模型已经确定了实体以及属性联系等，逻辑结构设计主要是针对 E-R 模型转化为关系模式，根据转化规则，逻辑模型设计结果如下，其中划线的属性为主码。

- 1. 用户（用户 ID，姓名，密码，激活状态，手机号，邮箱）
- 管理员（管理员 ID，姓名，密码，激活状态，手机号，邮箱，父管理员 ID）
- 房屋（房屋 ID、位置、房屋状态、管理员 ID）
- 投诉/建议（投诉/建议 ID、被投诉/建议的对象、时间、内容、管理员 ID，用户 ID）
- 消息（消息 ID、内容、时间、有效期、管理员 ID、消息类型）
- 费用（费用 ID、开始时间、缴费时间、金额、房屋 ID、用户 ID、费用类型 ID）
- 费用类型（费用类型 ID、费用类型）
- 维修任务（维修 ID、维修状态、内容、用户 ID）
- 停车位（停车位 ID、停车位位置、购买时间、停车位状态、房屋 ID）
- 用户\_房屋关联（用户 ID、房屋 ID、购房时间、入住时间）
- 房屋\_管理员关联（更新 ID、更新时间、更新内容、房屋 ID、管理员 ID）
- 管理员\_维修任务关联（维修记录 ID、维修状态、维修任务分配时间、维修完成时间、维修 ID、管理员 ID）

#### 3.2. 数据关系表结构

用户信息表(UserInfo)：用于记录用户的信息，手机号和邮箱必须一项非空且唯一，激活状态默认为未激活，主要数据结构如表 3-1 所示：

表 3-1 用户信息表(UserInfo)表结构

字段名	类型	长度	约束条件	备 注
UserID	int	11	自增，主键	用户 ID
UserName	varchar	12	非空	姓名
UserPassWord	varchar	128	非空	密码
IsActive	tinyint	1	非空，默认 0	是否激活（0：未激活，1：已激活）
UserPhone	varchar	11	唯一	手机号
UserEmail	varchar	20	唯一	邮箱

管理员信息表(AdminInfo)：用于记录管理员的信息，手机号和邮箱必须一项非空且唯一，激活状态默认为未激活，主要数据结构如表 3-2 所示：

表 3-2 管理员信息表(AdminInfo)表结构

字段名	类型	长度	约束条件	备 注
-----	----	----	------	-----

AdminID	int	11	自增，主键	管理员 ID
AdminName	varchar	12	非空	姓名
AdminPassWord	varchar	128	非空	密码
IsActive	tinyint	1	非空，默认 0	是否激活（0：未激活，1：已激活）
AdminPhone	varchar	11	唯一	手机号
AdminEmail	varchar	20	唯一	邮箱
PAdminID	int	11	默认 NULL	父管理员 ID

房屋信息表(HouseInfo)：用于记录房屋的基本信息，房屋状态有无用户和有用户两种状态，默认为无人状态。主要数据结构如表 3-3 所示：

表 3-3 房屋信息表(HouseInfo)表结构

字段名	类型	长度	约束条件	备 注
HouseID	int	11	自增，主键	房屋 ID
Location	varchar	20	非空	位置
HouseState	tinyint	1	非空，默认 0	房屋状态（0：无用户，1：有用户）
House_AdminID	int	11	非空，外键	管理员 ID

投诉/建议信息表(Complaint\_and\_AdviceInfo)：用于记录用户对小区各类服务的投诉/建议信息。主要数据结构如表 3-4 所示：

表 3-4 投诉/建议信息表(Complaint\_and\_AdviceInfo)表结构

字段名	类型	长度	约束条件	备 注
C_AID	int	11	自增，主键	投诉/建议 ID
C_ATime	Timestamp		非空，默认 CURRENT_TIMESTAMP	时间
C_AObject	varchar	20		对被投诉/建议的对象的描述
C_AContent	Text		非空	内容
C_A_UserID	int	11	非空，外键	用户 ID
C_A_AdminID	int	11	外键	管理员 ID

消息信息表(NewsInfo)：用于记录管理员发送的公告和信息，类型分为三类：单用户，多用户，全体用户。主要数据结构如表 3-5 所示：

表 3-5 消息信息表(NewsInfo)表结构

字段名	类型	长度	约束条件	备 注
NewsID	int	11	自增，主键	消息 ID
NewsTime	Timestamp		非空，默认 CURRENT_TIMESTAMP	发送时间
NewsContent	Text		非空	内容
News_AdminID	int	11	非空，外键	管理员 ID
NewsType	tinyint	1	非空，默认 0	消息类型（0：全体用户，1：多用户，2：单用户）
NewsEndTime	Timestamp			到期时间

费用类型信息表(CosttypeInfo)：用于记录各种费用类型信息。主要数据结构如表 3-6 所示：

表 3-6 费用类型信息表(CosttypeInfo)表结构

字段名	类型	长度	约束条件	备 注
CosttypeID	int	11	自增，主键	费用类型 ID
CostType	varchar	20	非空	费用类型

费用信息表(CostInfo)：用于记录各种费用信息。主要数据结构如表 3-7 所示：

表 3-7 费用信息表(CostInfo)表结构

字段名	类型	长度	约束条件	备 注
CostID	int	11	自增，主键	费用 ID
CostStartTime	Timestamp		默认 CURRENT_TIMESTAMP	开始时间
PayTime	Timestamp			缴费时间
Money	double	8	非空	金额
HouseID	int	11	非空，外键	房屋 ID
UserID	int	11	非空，外键	用户 ID

维修任务信息表(TaskInfo)：用于记录维修信息，维修状态分为（未开始，已开始，已完成）三种，默认为未开始。主要数据结构如表 3-8 所示：

表 3-8 维修任务信息表(TaskInfo)表结构

字段名	类型	长度	约束条件	备 注
TaskID	int	11	自增，主键	维修 ID

TaskState	tinyint	1	非空，默认 0	维修状态（0：未开始，1：已开始，2：已完成）
TaskContent	Text		非空	维修内容
Task_UserID	int	11	非空，外键	用户 ID

停车位信息表(PCarInfo)：用于记录停车位信息，状态分为三种（已购买，未购买）默认为空闲，只有当状态为已预定时，用户 ID 为非空。主要数据结构如表 3-9 所示：

表 3-9 停车位信息表(PCarInfo)表结构

字段名	类型	长度	约束条件	备 注
PCarID	int	11	自增，主键	停车 ID
PCarState	tinyint	1	非空，默认 0	停车位状态（1：已购买 0：未购买）
PCarLocation	varchar	20	非空	位置
BuyTime	Timestamp			购买时间
HouseID	int	11	外键	房屋 ID

用户\_房屋关联表(User\_HouseRelation)：用于记录房屋的归属用户，购买信息。主要数据结构如表 3-10 所示：

表 3-10 用户\_房屋关联表(User\_HouseRelation)表结构

字段名	类型	长度	约束条件	备 注
UH_Relation_UserID	int	11	外键，主键	用户 ID
UH_Relation_HouseID	int	11	外键，主键	房屋 ID
GetHouseTime	Timestamp		非空	购房时间
MoveTime	Timestamp		非空	入住时间

房屋\_管理员关联(UpdataRelation)：用于记录管理员更新房屋信息（一个管理员可能多次更新同一房屋信息）。主要数据结构如表 3-11 所示：

表 3-11 房屋\_管理员关联(UpdataRelation)表结构

字段名	类型	长度	约束条件	备 注
UpdataID	int	11	自增，主键	更新 ID
UpdataContent	Text		非空	更新内容



UpdataTime	Timestamp		非空，默认 CURRENT_TIMESTAMP	更新时间
Updata_HouseID	int	11	非空，外键	房屋 ID
Updata_AdminID	int	11	非空，外键	管理员 ID

管理员\_维修任务关联(Admin\_TaskRelation)：用于记录管理员分配维修任务信息，维修状态分为正在维修，和维修完成，维修失败三种，默认正在维修，当维修完成时，维修完成时间为非空。主要数据结构如表 3-12 所示：

表 3-12 管理员\_维修任务关联(Admin\_TaskRelation)表结构

字段名	类型	长度	约束条件	备 注
Admin_TaskID	int	11	自增，主键	维修信息 ID
ATaskState	tinyint	1	非空，默认 0	维修状态 (0：正在维修，1：维修完成，2：维修失败)
TaskStartTime	Timestamp		非空，默认 CURRENT_TIMESTAMP	维修任务分配时间
TaskEndTime	Timestamp			维修完成时间
Admin_Task_TaskID	int	11	非空，外键	维修 ID
Admin_Task_AdminID	int	11	非空，外键	管理员 ID

#### 4. 数据库及关系表的建立

##### 1. 创建数据库 (property\_management)

创建数据库 SQL 语句

```
Create database property_management;
```

##### 用户信息表 (userinfo)

(1) 创建表 SQL 语句

```
CREATE TABLE userinfo(  
    userid INT AUTO_INCREMENT PRIMARY KEY COMMENT '用户 ID',  
    username VARCHAR(12) NOT NULL COMMENT '用户名',  
    userpassword VARCHAR(255) NOT NULL COMMENT '密码',  
    isactive TINYINT NOT NULL COMMENT '是否激活 (0:未激活, 1:已激活)'  
DEFAULT (0),  
    userphone VARCHAR(11) UNIQUE COMMENT '手机号码',
```

useremail VARCHAR(50) UNIQUE COMMENT '邮箱'

);

(2) 产生表格式

mysql> DESC userinfo;

Field	Type	Null	Key	Default	Extra
userid	int	NO	PRI	NULL	auto_increment
username	varchar(12)	NO		NULL	
userpassword	varchar(255)	NO		NULL	
isactive	tinyint	NO		0	DEFAULT_GENERATED
userphone	varchar(11)	YES	UNI	NULL	
useremail	varchar(50)	YES	UNI	NULL	

6 rows in set (0.00 sec)

(3) 插入数据 SQL 语句

```
INSERT INTO userinfo (`userid`, `username`, `userpassword`, `isactive`,  
`userphone`, `useremail`) VALUES (1, '磨暮', 'jjfN8m$J(nJ9rX@auPwe', 1,  
'15362264107', 'wwmgbp@qq.com');
```

(4) 插入 10 条语句后的数据表

mysql> SELECT \* FROM userinfo LIMIT 10;

userid	username	userpassword	isactive	userphone	useremail
1	磨暮	jjfN8m\$J(nJ9rX@auPwe	1	15362264107	wwmgbp@qq.com;
2	介雅诗	7K1&pORh\$`vOr7Sc	1	15181495409	hobpgnesqkbjdn@sohu.com;
3	革清晖	d!TB03N(Q	1	13428002557	cvreqvbwj@35.com;
4	步金	D2dGT@XCd5T!7\$469	1	15351467466	efiogcpsdkr@35.com;
5	世寒凝	rU3kD(PcAZHz	1	15735257321	veb@35.com;
6	颜海荣	zav2tVcQ\$Spn7IW	1	18884415520	twjko@enet.com.cn;
7	占鸿德	avHZeX8y3&Z9k0	1	18560027894	vrjin@qq.com;
8	洧灵安	5oevHH1gcH	1	18592682871	opwm@263.net;
9	诸碧曼	q(`eEbUGJ)pACJ*	1	13514189910	rhlsikh@msn.com;
10	辛书双	sCdv+Btuy1&	1	18538232830	foutwmpmroelii@hotmail.com;

10 rows in set (0.00 sec)

管理员信息表(AdminInfo)

(1) 创建表 SQL 语句

CREATE TABLE admininfo(

adminid INT AUTO\_INCREMENT PRIMARY KEY COMMENT '管理员 ID',

adminname VARCHAR(12) NOT NULL COMMENT '管理员名',

adminpassword VARCHAR(255) NOT NULL COMMENT '密码',

isactive TINYINT NOT NULL COMMENT '是否激活 (0:未激活, 1:已激活)'

DEFAULT(0),

adminphone VARCHAR(11) UNIQUE COMMENT '手机号码',

adminemail VARCHAR(50) UNIQUE COMMENT '邮箱',

padminid INT DEFAULT(NULL) COMMENT '父管理员 ID',

FOREIGN KEY (padminid) REFERENCES admininfo(adminid)

);

(2) 产生表格式

```
mysql> DESC admininfo;
```

Field	Type	Null	Key	Default	Extra
adminid	int	NO	PRI	NULL	auto_increment
adminname	varchar(12)	NO		NULL	
adminpassword	varchar(255)	NO		NULL	
isactive	tinyint	NO		0	DEFAULT_GENERATED
adminphone	varchar(11)	YES	UNI	NULL	
adminemail	varchar(50)	YES	UNI	NULL	
padminid	int	YES	MUL	NULL	DEFAULT_GENERATED

```
7 rows in set (0.00 sec)
```

### (3) 插入数据 SQL 语句

```
INSERT INTO admininfo ('adminid', 'adminname', 'adminpassword',
'isactive', 'adminphone', 'adminemail', 'padminid') VALUES (21, '班代天',
'AZHR1p00XcXjy&k@$9\r\n', 1, '18414137037', 'thlrq@263.net', NULL),
('adminid', 'adminname', 'adminpassword', 'isactive', 'adminphone',
'adminemail', 'padminid') VALUES (24, '都凡之', 'pumusnDX(viM^sL', 1,
'13891263138', 'citwbdp@265.com', 21), ('adminid', 'adminname',
'adminpassword', 'isactive', 'adminphone', 'adminemail', 'padminid') VALUES
(2, '蒙舒方', '!ZBfutlizhT6m~TcB8le', 1, '15734882709', 'hdala@56.com',
21), ('adminid', 'adminname', 'adminpassword', 'isactive', 'adminphone',
'adminemail', 'padminid') VALUES (7, '戢齐敏', 'Wb&Hp!r%s%XhgsS&fDM', 1,
'18029828973', 'nibao@163.com', 21), ('adminid', 'adminname',
'adminpassword', 'isactive', 'adminphone', 'adminemail', 'padminid') VALUES
(15, '曾合美', '^g9&&ReWu4o', 1, '17377237638', 'wbu@163.net', 21);
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM admininfo LIMIT 10;
```

adminid	adminname	adminpassword	isactive	adminphone	adminemail	padminid
1	端才良	E+#bWPwIcrjRt~RBCET	1	13423874189	opjdpmraksskw@126.com	15
2	蒙舒方	!ZBfutlizhT6m~TcB8le	1	15734882709	hdala@56.com	21
3	竺碧蕊	ZP#TOW04cvmv3KX	1	14702202913	nutrhklpvjpefbp@qq.com	15
4	查书文	AJ`Wnq(aVWXQv9F	1	18889426395	aevfupwvdfvh@hotmail.com	15
5	占经略	PVuwkFxQQx	1	18109416374	apmtgsrkniruir@21cn.com	15
6	紫听然	EU6GJA`ns!	1	17203414523	goiflsavpeut@enet.com.cn	15
7	戢齐敏	Wb&Hp!r%s%XhgsS&fDM	1	18029828973	nibao@163.com	21
8	谭良朋	Tu(*dH9#HBpP	1	19879605630	fjvannmka@eastday.com	7
9	仇语海	%ZfoFeRVQTWR8&(	1	16639222254	dewgvov@163.net	7
10	褚运诚	ofLpaF!iXop	1	15274391432	oebvv@eastday.com	7

```
10 rows in set (0.00 sec)
```

## 2. 房屋信息表(HouseInfo)

### (1) 创建表 SQL 语句

```
CREATE TABLE houseinfo(
    houseid INT AUTO_INCREMENT PRIMARY KEY COMMENT '房屋 ID',
    location VARCHAR(20) NOT NULL COMMENT '位置',
    housestate TINYINT NOT NULL COMMENT '房屋状态(0:无用户, 1:有用户)'
    DEFAULT 0,
    house_adminid INT NOT NULL COMMENT '管理员 ID',
    FOREIGN KEY (house_adminid) REFERENCES admininfo(adminID)
```

);

### (2) 产生表格式

```
mysql> DESC houseinfo;
```

Field	Type	Null	Key	Default	Extra
houseid	int	NO	PRI	NULL	auto_increment
location	varchar(20)	NO		NULL	
housestate	tinyint	NO		0	
house_adminid	int	NO	MUL	NULL	

4 rows in set (0.00 sec)

### (3) 插入数据 SQL 语句

```
INSERT INTO houseinfo (`houseid`, `location`, `housestate`,  
`house_adminid`) VALUES (1, '清溪小区 4 栋 8 楼 20 号', 1, 8), (2, '清溪小区 5  
栋 2 楼 12 号', 1, 16);
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM houseinfo LIMIT 10;
```

houseid	location	housestate	house_adminid
1	清溪小区4 栋8楼20号	1	8
2	清溪小区5 栋2楼12号	1	16
3	清溪小区4 栋4楼11号	1	25
4	清溪小区3 栋11楼8号	1	26
5	清溪小区5 栋11楼17号	1	13
6	清溪小区1 栋10楼10号	1	1
7	清溪小区3 栋2楼16号	1	1
8	清溪小区5 栋5楼13号	1	20
9	清溪小区3 栋12楼13号	1	28
10	清溪小区4 栋2楼19号	1	23

10 rows in set (0.00 sec)

## 3. 投诉/建议信息表(Complaint\_and\_AdviceInfo)

### (1) 创建表 SQL 语句

```
CREATE TABLE complaint_and_acviceinfo(  
    c_aid INT AUTO_INCREMENT PRIMARY KEY COMMENT '投诉/建议 ID',  
    c_atime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '时间',  
    c_aobject VARCHAR(20) COMMENT '对被投诉/建议的对象的描述',  
    c_acontent TEXT NOT NULL COMMENT '内容',  
    c_a_userid INT NOT NULL COMMENT '用户 ID',  
    c_a_adminid INT COMMENT '管理员 ID',  
    FOREIGN KEY (c_a_userid) REFERENCES userinfo(userid),  
    FOREIGN KEY (c_a_adminid) REFERENCES admininfo(adminID)  
);
```

### (2) 产生表格式

```
mysql> DESC complaint_and_acviceinfo;
```

Field	Type	Null	Key	Default	Extra
c_aid	int	NO	PRI	NULL	auto_increment
c_atime	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED
c_aobject	varchar(20)	YES		NULL	
c_acontent	text	NO		NULL	
c_a_userid	int	NO	MUL	NULL	
c_a_adminid	int	YES	MUL	NULL	

6 rows in set (0.00 sec)

### (3) 插入数据 SQL 语句

```
INSERT INTO complaint_and_acviceinfo (`c_aid`, `c_atime`, `c_aobject`,
`c_acontent`, `c_a_userid`, `c_a_adminid`) VALUES (1, '2018-03-21
06:25:59', 'xxxxxx', 'xxxxxxx', 14, 20), (2, '2017-11-07 09:23:03',
'xxxxxx', 'xxxxxxx', 55, 25);
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM complaint_and_acviceinfo LIMIT 10;
```

c_aid	c_atime	c_aobject	c_acontent	c_a_userid	c_a_adminid
1	2018-03-21 06:25:59	xxxxxx	xxxxxxx	9	20
2	2017-11-07 09:23:03	xxxxxx	xxxxxxx	19	25
3	2019-01-10 16:07:20	xxxxxx	xxxxxxx	55	11
4	2018-02-16 13:01:35	xxxxxx	xxxxxxx	80	20
5	2020-05-24 16:10:30	xxxxxx	xxxxxxx	67	20
6	2018-08-18 17:34:55	xxxxxx	xxxxxxx	4	1
7	2021-05-21 22:25:38	xxxxxx	xxxxxxx	43	11
8	2021-09-06 02:20:29	xxxxxx	xxxxxxx	62	1
9	2021-10-03 23:58:34	xxxxxx	xxxxxxx	87	17
10	2017-07-12 09:47:10	xxxxxx	xxxxxxx	38	3

10 rows in set (0.00 sec)

## 4. 消息信息表 (NewsInfo)

### (1) 创建表 SQL 语句

```
CREATE TABLE newsinfo(
    newsid INT AUTO_INCREMENT PRIMARY KEY COMMENT '消息 ID',
    newstime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '发送
时间',
    newscontent TEXT NOT NULL COMMENT '内容',
    newstype TINYINT NOT NULL DEFAULT 0 COMMENT '房屋状态(0:全体用户,
1:多用户, 2:单用户)',
    newsendtime TIMESTAMP COMMENT '到期时间',
    news_adminid INT NOT NULL COMMENT '管理员 ID',
    FOREIGN KEY (news_adminid) REFERENCES admininfo(adminid)
);
```

### (2) 产生表格式

```
mysql> DESC newsinfo;
```

Field	Type	Null	Key	Default	Extra
newsid	int	NO	PRI	NULL	auto_increment
newstime	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED
newscontent	text	NO		NULL	
newstype	tinyint	NO		0	
newsendtime	timestamp	YES		NULL	
news_adminid	int	NO	MUL	NULL	

6 rows in set (0.00 sec)

### (3) 插入数据 SQL 语句

```
INSERT INTO newsinfo (`newsid`, `newstime`, `newscontent`,
`news_adminid`, `newstype`, `newsendtime`) VALUES (1, '2020-12-16
08:37:21', 'xxx', 20, 1, '2020-04-18 07:03:44'), (2, '2019-05-19 11:59:08',
'xxx', 25, 2, NULL);
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM newsinfo LIMIT 10;
```

newsid	newstime	newscontent	newstype	newsendtime	news_adminid
1	2020-12-16 08:37:21	xxx	1	2020-04-18 07:03:44	20
2	2019-05-19 11:59:08	xxx	2	NULL	25
3	2018-10-28 15:53:00	xxx	2	NULL	11
4	2022-04-17 06:51:02	xxx	0	2019-03-10 00:17:37	20
5	2018-03-20 13:07:54	xxx	1	2019-08-14 21:58:41	20
6	2017-09-18 03:05:31	xxx	2	NULL	1
7	2021-05-03 05:03:52	xxx	1	2020-02-19 07:52:08	11
8	2020-10-17 08:42:40	xxx	1	2019-06-17 07:57:21	1
9	2020-09-23 05:26:47	xxx	2	NULL	17
10	2017-03-19 23:06:12	xxx	0	2018-10-20 12:11:09	3

10 rows in set (0.00 sec)

## 5. 费用类型信息表 (CosttypeInfo)

### (1) 创建表 SQL 语句

```
CREATE TABLE costtypeinfo(
    costtypeid int AUTO_INCREMENT PRIMARY KEY COMMENT '费用类型 ID',
    costtype VARCHAR(20) NOT NULL COMMENT '费用类型'
);
```

### (2) 产生表格式

```
mysql> DESC costtypeinfo;
```

Field	Type	Null	Key	Default	Extra
costtypeid	int	NO	PRI	NULL	auto_increment
costtype	varchar(20)	NO		NULL	

2 rows in set (0.00 sec)

### (3) 插入数据 SQL 语句

```
INSERT INTO costtypeinfo (`costtypeid`, `costtype`) VALUES (1, '水费'), (2, '电费');
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM costtypeinfo LIMIT 6;
```

costtypeid	costtype
1	水费
2	电费
3	气费
4	维修费
5	物管费
6	清洁费

6 rows in set (0.00 sec)

## 6. 费用信息表(CostInfo)

### (1) 创建表 SQL 语句

```
CREATE TABLE costinfo(
    costid INT AUTO_INCREMENT PRIMARY KEY COMMENT '费用 ID',
    coststarttime TIMESTAMP DEFAULT CURRENT_TIMESTAMP COMMENT '开始时间',
    paytime TIMESTAMP COMMENT '缴费时间',
    costmoney DOUBLE NOT NULL COMMENT '金额',
    cost_houseid INT NOT NULL COMMENT '房屋 ID',
    cost_userid INT NOT NULL COMMENT '用户 ID',
    cost_costtypeid INT NOT NULL COMMENT '费用类型 ID',
    FOREIGN KEY (cost_houseid) REFERENCES houseinfo(houseid),
    FOREIGN KEY (cost_userid) REFERENCES userinfo(userid),
    FOREIGN KEY (cost_costtypeid) REFERENCES costtypeinfo(costtypeid)
);
```

### (2) 产生表格式

```
mysql> DESC costinfo;
```

Field	Type	Null	Key	Default	Extra
costid	int	NO	PRI	NULL	auto_increment
coststarttime	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
paytime	timestamp	YES		NULL	
costmoney	double	NO		NULL	
cost_houseid	int	NO	MUL	NULL	
cost_userid	int	NO	MUL	NULL	
cost_costtypeid	int	NO	MUL	NULL	

7 rows in set (0.00 sec)

### (3) 插入数据 SQL 语句

```
INSERT INTO costinfo (`costid`, `coststarttime`, `paytime`,
`costmoney`, `cost_userid`, `cost_houseid`, `cost_costtypeid`) VALUES (1,
'2021-06-08 12:33:55', '2021-06-13 07:43:55', 170.083920017742514, 7, 20,
2), (2, '2020-09-15 08:09:39', '2020-09-16 10:03:03', 636.686875427754899,
36, 7, 3);
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM costinfo LIMIT 10;
```

costid	coststarttime	paytime	costmoney	cost_houseid	cost_userid	cost_costtypeid
1	2021-06-08 12:33:55	2021-06-13 07:43:55	232.78487872169728	20	7	5
2	2020-09-15 08:09:39	2020-09-16 10:03:03	468.67646915333125	7	36	6
3	2019-07-31 06:36:49	2019-08-02 12:45:22	329.82261341522945	26	41	5
4	2018-01-12 18:46:59	2018-01-17 11:39:13	932.2156533715017	24	48	2
5	2017-05-16 17:22:13	2017-05-20 02:35:06	563.5173007753338	39	66	2
6	2017-11-05 00:31:37	2017-11-07 10:30:27	499.53757386691956	13	90	6
7	2019-01-30 04:35:13	2019-02-03 16:52:05	232.07654067646848	8	10	6
8	2017-01-16 05:45:53	2017-01-21 17:27:24	140.75144300916793	11	34	2
9	2019-03-13 06:39:08	2019-03-13 20:50:08	67.16266014073472	31	79	2
10	2021-12-02 21:53:48	2021-12-06 04:47:59	574.4748885431089	27	39	6

10 rows in set (0.00 sec)

## 7. 维修任务信息表(TaskInfo)

### (1) 创建表 SQL 语句

```
CREATE TABLE taskinfo(
```

```
    taskid INT AUTO_INCREMENT PRIMARY KEY COMMENT '维修 ID',
```

```
    taskstate TINYINT NOT NULL DEFAULT 0 COMMENT '维修状态(0:未开始, 1:已开始, 2:已完成)',
```

```
    taskcontent TEXT NOT NULL COMMENT '维修内容',
```

```
    task_userid INT NOT NULL COMMENT '用户 ID',
```

```
    FOREIGN KEY (task_userid) REFERENCES userinfo(userid)
```

```
);
```

### (2) 产生表格式

```
mysql> DESC taskinfo;
```

Field	Type	Null	Key	Default	Extra
taskid	int	NO	PRI	NULL	auto_increment
taskstate	tinyint	NO		0	
taskcontent	text	NO		NULL	
task_userid	int	NO	MUL	NULL	

4 rows in set (0.00 sec)

### (3) 插入数据 SQL 语句

```
INSERT INTO taskinfo (`taskid`, `taskstate`, `taskcontent`,  
`task_userid`) VALUES (1, 2, '暖气故障', 70), (2, 2, '修门', 77);
```

### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM taskinfo LIMIT 10;
```

taskid	taskstate	taskcontent	task_userid
1	2	暖气故障	83
2	1	电路维修	57
3	1	配玻璃	73
4	2	修门	2
5	1	通下水道	95
6	2	涂油漆	7
7	0	水管破裂	14
8	0	修补屋面	63
9	2	暖气故障	31
10	2	暖气故障	78

10 rows in set (0.00 sec)

## 8. 停车位信息表(PCarInfo)



(1) 创建表 SQL 语句

```
CREATE TABLE pcarinfo(  
    pcarid INT AUTO_INCREMENT PRIMARY KEY COMMENT '停车位 ID',  
    pcarstate TINYINT NOT NULL DEFAULT 0 COMMENT '停车位状态(0:空闲,  
1:已停车, 2:已预定)',  
    pcarloction VARCHAR(20) NOT NULL COMMENT '位置',  
    buytime TIMESTAMP COMMENT '购买时间',  
    pcar_houseid INT COMMENT '房屋 ID',  
    FOREIGN KEY (pcar_houseid) REFERENCES houseinfo(houseid)  
);
```

(2) 产生表格式

```
mysql> DESC pcarinfo;
```

Field	Type	Null	Key	Default	Extra
pcarid	int	NO	PRI	NULL	auto_increment
pcarstate	tinyint	NO		0	
pcarloction	varchar(20)	NO		NULL	
buytime	timestamp	YES		NULL	
pcar_houseid	int	YES	MUL	NULL	

```
5 rows in set (0.00 sec)
```

(3) 插入数据 SQL 语句

```
INSERT INTO pcarinfo (`pcarid`, `pcarstate`, `pcarloction`, `buytime`,  
`pcar_houseid`) VALUES (1, 1, '1', '2017-03-26 17:31:07', 26), (2, 1, '2',  
'2021-04-19 05:34:36', 22);
```

(4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM pcarinfo LIMIT 10;
```

pcarid	pcarstate	pcarloction	buytime	pcar_houseid
1	1	1	2017-03-26 17:31:07	26
2	1	2	2021-04-19 05:34:36	22
3	1	3	2017-09-09 15:56:29	13
4	0	4	NULL	12
5	1	5	2021-08-04 23:41:45	9
6	1	6	2019-01-30 14:56:23	35
7	1	7	2017-03-23 10:59:06	30
8	1	8	2018-07-22 04:18:52	26
9	0	9	NULL	10
10	1	10	2017-09-24 12:00:11	2

```
10 rows in set (0.00 sec)
```

9. 用户\_房屋关联表(User\_HouseRelation)

(1) 创建表 SQL 语句

```
CREATE TABLE user_houserelation(  
    uh_relation_userid INT COMMENT '用户 ID',  
    uh_relation_houseid INT COMMENT '房屋 ID',  
    gethousetime TIMESTAMP COMMENT '购房时间',
```

```

movetime TIMESTAMP COMMENT '入住时间',
PRIMARY KEY (uh_relation_userid, uh_relation_houseid),
FOREIGN KEY (uh_relation_userid) REFERENCES userinfo(userid),
FOREIGN KEY (uh_relation_houseid) REFERENCES houseinfo(houseid)
);

```

#### (2) 产生表格式

```
mysql> DESC user_houserelation;
```

Field	Type	Null	Key	Default	Extra
uh_relation_userid	int	NO	PRI	NULL	
uh_relation_houseid	int	NO	PRI	NULL	
gethousetime	timestamp	YES		NULL	
movetime	timestamp	YES		NULL	

```
4 rows in set (0.00 sec)
```

#### (3) 插入数据 SQL 语句

```

INSERT INTO user_houserelation (`uh_relation_userid`,
`uh_relation_houseid`, `gethousetime`, `movetime`) VALUES (30, 40, '2018-
07-25 08:54:34', '2018-10-13 03:20:08'), (64, 40, '2017-12-13 01:51:14',
'2018-11-25 11:01:23');

```

#### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM user_houserelation LIMIT 10;
```

uh_relation_userid	uh_relation_houseid	gethousetime	movetime
1	33	2019-09-23 04:47:23	2020-06-07 14:59:05
2	12	2019-04-05 01:18:57	2019-12-15 08:24:48
4	39	2020-12-17 01:31:08	2021-03-05 05:19:10
5	5	2018-01-14 04:11:03	2018-11-11 20:12:33
6	30	2022-01-12 10:29:50	2022-12-04 21:53:31
7	20	2019-03-24 04:34:15	2020-01-06 07:09:57
8	35	2018-06-05 15:50:17	2019-05-21 08:16:23
9	37	2019-03-22 12:45:48	2020-01-27 05:20:33
10	8	2019-01-19 18:58:05	2019-03-30 12:38:58
11	29	2019-01-01 21:39:21	2019-06-20 07:13:22

```
10 rows in set (0.00 sec)
```

### 10. 房屋\_管理员关联(UpdataRelation)

#### (1) 创建表 SQL 语句

```
CREATE TABLE updatarelation(
```

```

    updataid INT AUTO_INCREMENT PRIMARY KEY COMMENT '更新 ID',

```

```

    updatacontent TEXT NOT NULL COMMENT '更新内容',

```

```

    updatatime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '更

```

新时间’,

```
updata_houseid INT NOT NULL COMMENT '房屋 ID',
updata_adminid INT NOT NULL COMMENT '管理员 ID',
FOREIGN KEY (updata_houseid) REFERENCES houseinfo(houseid) ,
FOREIGN KEY (updata_adminid) REFERENCES admininfo(adminid)
);
```

#### (2) 产生表格式

```
mysql> DESC updatarelation;
```

Field	Type	Null	Key	Default	Extra
updataid	int	NO	PRI	NULL	auto_increment
updatacontent	text	NO		NULL	
updatatime	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED
updata_houseid	int	NO	MUL	NULL	
updata_adminid	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

#### (3) 插入数据 SQL 语句

```
INSERT INTO updatarelation (`updataid`, `updatacontent`, `updatatime`,
`updata_houseid`, `updata_adminid`) VALUES (1, 'xxxx', '2018-01-17
06:07:09', 39, 8), (2, 'xxxx', '2017-06-19 12:07:37', 13, 8);
```

#### (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM updatarelation LIMIT 10;
```

updataid	updatacontent	updatatime	updata_houseid	updata_adminid
1	xxxx	2018-01-17 06:07:09	39	8
2	xxxx	2017-06-19 12:07:37	13	8
3	xxxx	2021-09-21 04:16:07	24	9
4	xxxx	2019-08-11 09:08:38	34	20
5	xxxx	2018-09-16 23:50:45	12	17
6	xxxx	2017-01-13 23:41:31	11	4
7	xxxx	2022-04-17 02:16:48	37	16
8	xxxx	2022-02-24 11:20:05	4	19
9	xxxx	2017-04-09 09:30:42	35	14
10	xxxx	2017-11-15 01:17:42	8	20

10 rows in set (0.00 sec)

### 11. 管理员\_维修任务关联(Admin\_TaskRelation)

#### (1) 创建表 SQL 语句

```
CREATE TABLE admin_taskrelation(
    admin_taskid INT AUTO_INCREMENT PRIMARY KEY COMMENT '维修信息',
    ataskstate TINYINT NOT NULL DEFAULT 0 COMMENT '维修状态(0:正在维修,
1:维修完成, 2:维修失败)',
    taskstarttime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT
'维修任务分配时间',
    taskendtime TIMESTAMP COMMENT '维修完成时间',
    admin_task_taskid INT NOT NULL COMMENT '维修 ID',
    admin_task_adminid INT NOT NULL COMMENT '管理员 ID',
    FOREIGN KEY (admin_task_taskid) REFERENCES taskinfo(taskid),
```

```
FOREIGN KEY (admin_task_adminid) REFERENCES admininfo(adminid)
);
```

## (2) 产生表格式

```
mysql> DESC admin_taskrelation;
```

Field	Type	Null	Key	Default	Extra
admin_taskid	int	NO	PRI	NULL	auto_increment
ataskstate	tinyint	NO		0	
taskstarttime	timestamp	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED
taskendtime	timestamp	YES		NULL	
admin_task_taskid	int	NO	MUL	NULL	
admin_task_adminid	int	NO	MUL	NULL	

```
6 rows in set (0.00 sec)
```

## (3) 插入数据 SQL 语句

```
INSERT INTO admin_taskrelation (`admin_taskid`, `ataskstate`,
`taskstarttime`, `taskendtime`, `admin_task_taskid`, `admin_task_adminid`)
VALUES (67, 1, '2020-02-23 21:48:25', '2017-12-11 18:03:04', 61, 13), (68,
1, '2018-04-02 03:22:27', '2021-02-21 13:53:33', 62, 13);
```

## (4) 插入 10 条语句后的数据表

```
mysql> SELECT * FROM admin_taskrelation LIMIT 10;
```

admin_taskid	ataskstate	taskstarttime	taskendtime	admin_task_taskid	admin_task_adminid
1	2	2020-01-24 17:39:23	NULL	98	11
2	2	2019-11-17 16:14:02	NULL	29	4
3	2	2021-11-11 22:25:17	NULL	48	18
4	2	2020-12-23 05:42:23	NULL	35	6
5	2	2020-12-29 15:04:01	NULL	6	13
6	2	2019-09-03 12:09:25	NULL	54	25
7	1	2020-09-17 07:55:23	2019-03-06 15:25:34	1	23
8	1	2018-01-12 04:43:12	2018-06-29 23:15:09	2	8
9	1	2018-04-26 11:44:30	2019-08-15 20:48:57	3	1
10	1	2017-10-20 02:02:32	2017-08-01 05:42:02	4	20

```
10 rows in set (0.00 sec)
```

附 原 始 实 验 记 录	<p><b>四、实验总结</b></p> <p>（1）用关系数据库理论对自己设计的数据模型进行评价，指出合理和不足之处，提出改进的方案。</p> <p>数据库的基本结构分三个层次，即物理数据层、概念数据层、逻辑数据层。</p> <p>在概念数据层和逻辑数据层：及构建 E—R 模型和建立数据库时，经过小组讨论和老师修正已具有较强的合理性，在现在的模式下大大减小了数据的冗余度。</p> <p>不知道能不能很好的契合后续的开发</p> <p>改进方案：在以后的开发过程中对数据库进行合理的修改</p> <p>（2）在关系定义和数据装载过程中遇到了哪些问题，是怎么解决的。</p> <p>1. 在创建管理员信息表时外键问题：</p> <p>解决方法：在创建表时，先添加父管理员，在添加子管理员。</p> <p>2.：导入数据时因为外键导致导入失败</p> <p>解决方法：忽略外键。</p>
---------------------------------	---