本系统所需的表：

1. 单元房表 room .
2. 业主表 owners .
3. 管理员表 admin .
4. 投诉表 complain .
5. 建议表 suggest .
6. 公告表 notice .
7. 报修表 upkeep .
8. 缴费表 pay .
9. 聊天表 chat .
10. 活动表 activity .
11. 参与表 joins .
12. 活动审核表 verify

数据字典：

1、**单元房表**：（已缴费：YES；未缴费：NO）

（ 房间ID（**room\_id** int），

房间大小（room\_size int），

是否缴费（room\_pay varchar（3）），

业主ID（room\_owner int））

主键：**room\_id**

|  |  |
| --- | --- |
| Name | Code |
| 房间ID | room\_id |
| 房间大小 | room\_size |
| 是否缴费 | room\_pay |
| 业主ID | room\_owner |

CREATE TABLE room (

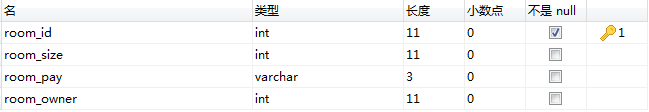
room\_id INT PRIMARY KEY NOT NULL auto\_increment,

room\_size INT,

room\_pay VARCHAR (3),

room\_owner INT

);



2、**业主表**：

（ 业主ID（**owners\_id**  int），

邮箱（owners\_email varchar（50）），

房间ID（room\_id int），

密码（owners\_password varchar（50）），

姓名（owners\_name varchar（50）），

性别（owners\_sex varchar（2）），

电话号码（owners\_phone varchar（50）））

主键：**owners\_id** 外键：room\_id

|  |  |
| --- | --- |
| Name | Code |
| 业主ID | owners\_id |
| 邮箱 | owners\_email |
| 房间ID | room\_id |
| 密码 | owners\_password |
| 姓名 | owners\_name |
| 性别 | owners\_sex |
| 电话号码 | owners\_phone |

CREATE TABLE owners (

owners\_id INT PRIMARY KEY NOT NULL auto\_increment,

owners\_email VARCHAR (50),

owners\_password VARCHAR (50),

owners\_name VARCHAR (50),

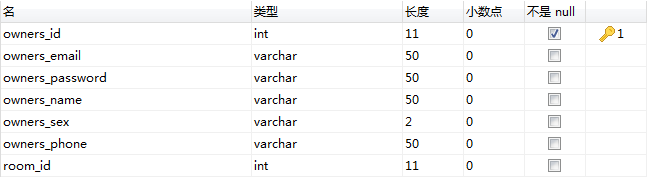
owners\_sex VARCHAR (2) DEFAULT '男',

owners\_phone VARCHAR (50),

room\_id INT,

FOREIGN KEY (room\_id) REFERENCES room (room\_id)

);



3、**管理员表**：

（ 管理员ID（**admin\_id**  int），

账号（admin\_account varchar（50）），

密码（admin\_password varchar（50）），

姓名（admin\_name varchar（50）），

电话（admin\_phone varchar（50）））

主键：**admin\_id**

|  |  |
| --- | --- |
| Name | Code |
| 管理员ID | admin\_id |
| 账号 | admin\_account |
| 密码 | admin\_password |
| 姓名 | admin\_name |
| 电话 | admin\_phone |

CREATE TABLE admin (

admin\_id INT PRIMARY KEY NOT NULL auto\_increment,

admin\_account VARCHAR (50),

admin\_password VARCHAR (50),

admin\_name VARCHAR (50),

admin\_phone VARCHAR (50)

);



4、**投诉表**：

（ 投诉ID（**complain\_id** int），

业主ID（owners\_id int），

投诉内容（complain\_content varchar（500）），

投诉时间（complain\_time datetime），

处理时间（complain\_handle\_time），

处理结果（complain\_handle\_result varchar（500）），

管理员ID（admin\_id int）

主键：**complain\_id** 外键：owners\_id、admin\_id

|  |  |
| --- | --- |
| Name | Code |
| 投诉ID | complain\_id |
| 业主ID | owners\_id |
| 投诉内容 | complain\_content |
| 投诉时间 | complain\_time |
| 处理时间 | complain\_handle\_time |
| 处理结果 | complain\_handle\_result |
| 管理员ID | admin\_id |

CREATE TABLE complain (

complain\_id INT PRIMARY KEY NOT NULL auto\_increment,

complain\_content VARCHAR (500),

complain\_time datetime,

complain\_handle\_time datetime,

complain\_handle\_result VARCHAR (500),

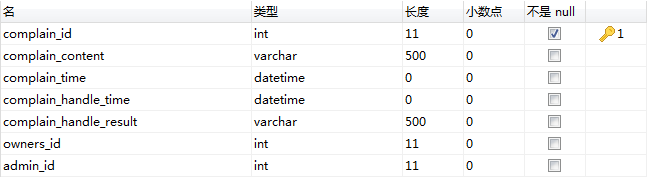
owners\_id INT,

admin\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id),

FOREIGN KEY (admin\_id) REFERENCES admin (admin\_id)

);



5、**建议表**：

（ 建议ID（**suggest\_id** int），

业主ID（owners\_id int），

建议内容（suggest\_content varchar（500）），

建议时间（suggest\_time datetime），

处理时间（suggest\_handle\_time datetime），

处理结果（suggest\_handle\_result varchar（500）），

管理员ID（admin\_id int）

主键：**suggest\_id** 外键：owners\_id、admin\_id

|  |  |
| --- | --- |
| Name | Code |
| 建议ID | suggest\_id |
| 业主ID | owners\_id |
| 建议内容 | suggest\_content |
| 建议时间 | suggest\_time |
| 处理时间 | suggest\_handle\_time |
| 处理结果 | suggest\_handle\_result |
| 管理员ID | admin\_id |

CREATE TABLE suggest (

suggest\_id INT PRIMARY KEY NOT NULL auto\_increment,

suggest\_content VARCHAR (500),

suggest\_time datetime,

suggest\_handle\_time datetime,

suggest\_handle\_result VARCHAR (500),

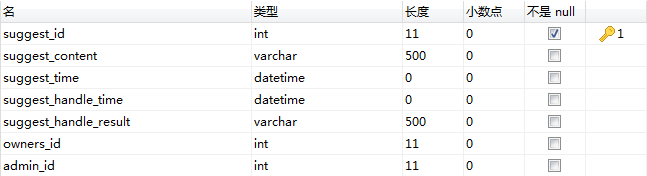
owners\_id INT,

admin\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id),

FOREIGN KEY (admin\_id) REFERENCES admin (admin\_id)

);



6、**公告表**：

（ 公告ID（**notice\_id** int），

公告标题（notice\_title varchar(500)），

内容（notice\_content varchar（500）），

发布时间（notice\_time datetime），

管理员ID（admin\_id int）

管理员姓名（admin\_name varchar(50)））

主键：**notice\_id** 外键：admin\_id

|  |  |
| --- | --- |
| Name | Code |
| 公告ID | notice\_id |
| 公告标题 | notice\_title |
| 公告内容 | notice\_content |
| 公告发布时间 | notice\_time |
| 管理员ID | admin\_id |
| 管理员姓名 | admin\_name |

CREATE TABLE notice (

notice\_id INT PRIMARY KEY NOT NULL auto\_increment,

notice\_title VARCHAR (100),

notice\_content VARCHAR (500),

notice\_time datetime,

admin\_id INT,

FOREIGN KEY (admin\_id) REFERENCES admin (admin\_id)

);

7、**报修表**：

（ 报修ID（**upkeep\_id** int），

业主ID（owners\_id int），

报修内容（upkeep\_content varchar（500）），

报修时间（upkeep\_time datetime），

处理时间（upkeep\_handle\_time datetime），

处理过程（upkeep\_process varchar（500）），

处理结果（upkeep\_result varchar（500）），

管理员ID（处理人）（admin\_id int）

主键：**upkeep\_id** 外键：owners\_id、admin\_id

|  |  |
| --- | --- |
| Name | Code |
| 报修ID | upkeep\_id |
| 业主ID | owners\_id |
| 报修内容 | upkeep\_content |
| 报修时间 | upkeep\_time |
| 处理时间 | upkeep\_handle\_time |
| 处理过程 | upkeep\_process |
| 处理结果 | upkeep\_result |
| 管理员ID | admin\_id |

CREATE TABLE upkeep (

upkeep\_id INT PRIMARY KEY NOT NULL auto\_increment,

upkeep\_content VARCHAR (500),

upkeep\_time datetime,

upkeep\_handle\_time datetime,

upkeep\_process VARCHAR (500),

upkeep\_result VARCHAR (500),

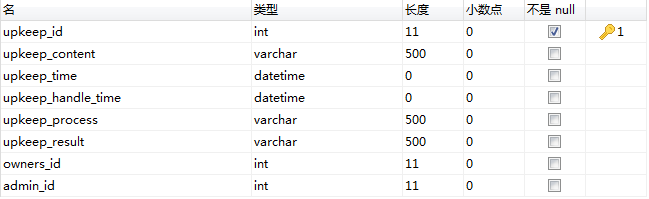
owners\_id INT,

admin\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id),

FOREIGN KEY (admin\_id) REFERENCES admin (admin\_id)

);



8、**缴费表**：

（ 缴费ID（**pay\_id** int），

业主ID（owners\_id int），

缴费时间（pay\_time datetime），

缴费金额（pay\_price double））

主键：**pay\_id** 外键：owners\_id

|  |  |
| --- | --- |
| Name | Code |
| 缴费ID | pay\_id |
| 业主ID | owners\_id |
| 缴费时间 | pay\_time |
| 缴费金额 | pay\_price |

CREATE TABLE pay (

pay\_id INT PRIMARY KEY NOT NULL auto\_increment,

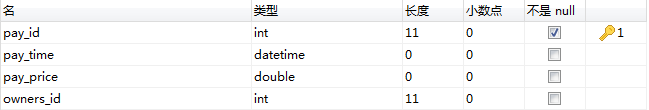
pay\_time datetime,

pay\_price DOUBLE,

owners\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id)

);



9、**聊天表**：

（ 聊天ID（**chat\_id** int），

聊天内容（chat\_content varchar（500）），

业主ID（发表人）（owners\_id int），

时间（chat\_time datetime）

主键：**chat\_id** 外键：owners\_id

|  |  |
| --- | --- |
| Name | Code |
| 聊天ID | chat\_id |
| 聊天内容 | chat\_content |
| 业主ID（发表人） | owners\_id |
| 时间 | chat\_time |

CREATE TABLE chat (

chat\_id INT PRIMARY KEY NOT NULL auto\_increment,

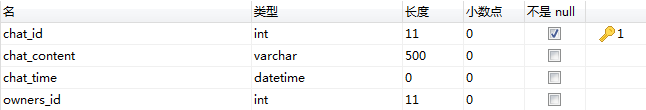
chat\_content VARCHAR (500),

chat\_time datetime,

owners\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id)

);



10、**活动表**：

（ 活动ID（**activity\_id** int），

活动内容（activity\_content varchar（500）），

活动时间（activity\_time datetime），

业主ID（发起人）（owners\_id int））

主键：**activity\_id** 外键：owners\_id

|  |  |
| --- | --- |
| Name | Code |
| 活动ID | activity\_id |
| 活动内容 | activity\_content |
| 活动时间 | activity\_time |
| 业主ID（发起人） | owners\_id |

CREATE TABLE activity (

activity\_id INT PRIMARY KEY NOT NULL auto\_increment,

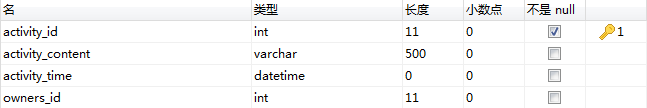
activity\_content VARCHAR (500),

activity\_time datetime,

owners\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id)

);



11、**参与表**：（用来连接活动表和用户表，参加人员）

（ 参与ID（**joins\_id** int），

活动ID（activity\_id int），

业主ID（参与人）（owners\_id int））

主键：joins\_id 外键：activity\_id、owners\_id

|  |  |
| --- | --- |
| 参与ID | joins\_id |
| 活动ID | activity\_id |
| 业主ID（参与人） | owners\_id |

CREATE TABLE joins (

joins\_id INT PRIMARY KEY NOT NULL auto\_increment,

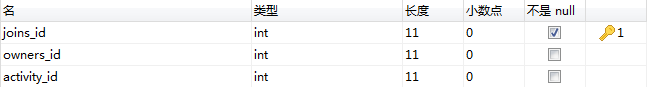
owners\_id INT,

activity\_id INT,

FOREIGN KEY (owners\_id) REFERENCES owners (owners\_id),

FOREIGN KEY (activity\_id) REFERENCES activity (activity\_id)

);



1. **活动审核表**：（结果用**YES**：过； **NO**：不过）

（ 审核ID（**verify\_id** int），

活动ID（activity\_id int），

审核内容（verify\_content varchar（500）），

审核结果（verify\_result varchar（3）），

审核时间（verify\_time datetime），

管理员ID（审核人）（admin\_id int））

主键：**verify\_id** 外键：activity\_id、admin\_id

|  |  |
| --- | --- |
| 审核ID | verify\_id |
| 活动ID | activity\_id |
| 审核内容 | verify\_content |
| 审核结果 | verify\_result |
| 审核时间 | verify\_time |
| 管理员ID（审核人） | admin\_id |

CREATE TABLE verify (

verify\_id INT PRIMARY KEY NOT NULL auto\_increment,

verify\_content VARCHAR (500),

verify\_result VARCHAR (3) DEFAULT 'NO',

verify\_time datetime,

activity\_id INT,

admin\_id INT,

FOREIGN KEY (activity\_id) REFERENCES activity (activity\_id),

FOREIGN KEY (admin\_id) REFERENCES admin (admin\_id)

);

