

# 数据库PJ说明文档

By 18302010034黄子豪

## 数据库PJ说明文档

[数据库设计ER图](#)

[表结构说明](#)

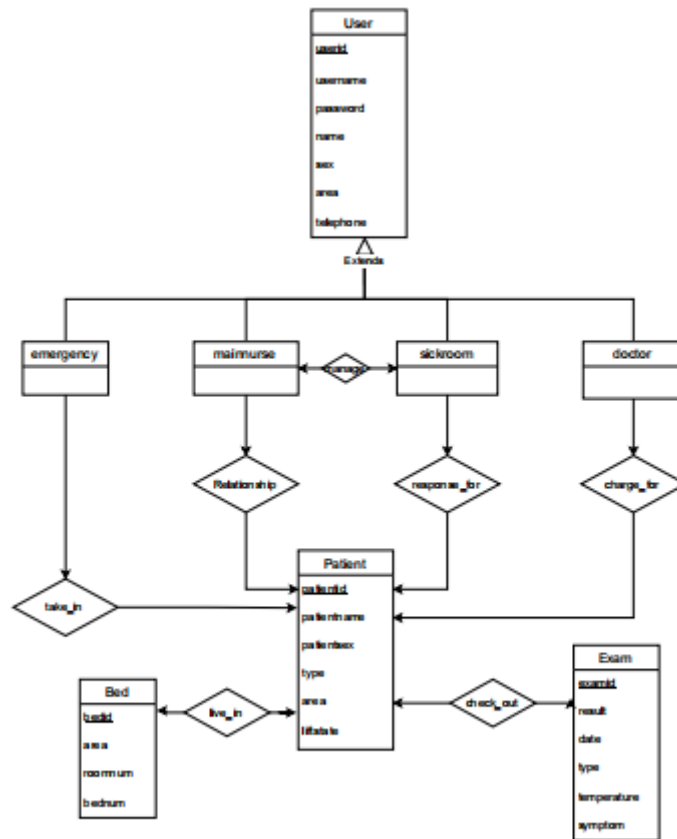
[索引定义说明](#)

[核心功能的SQL语句](#)

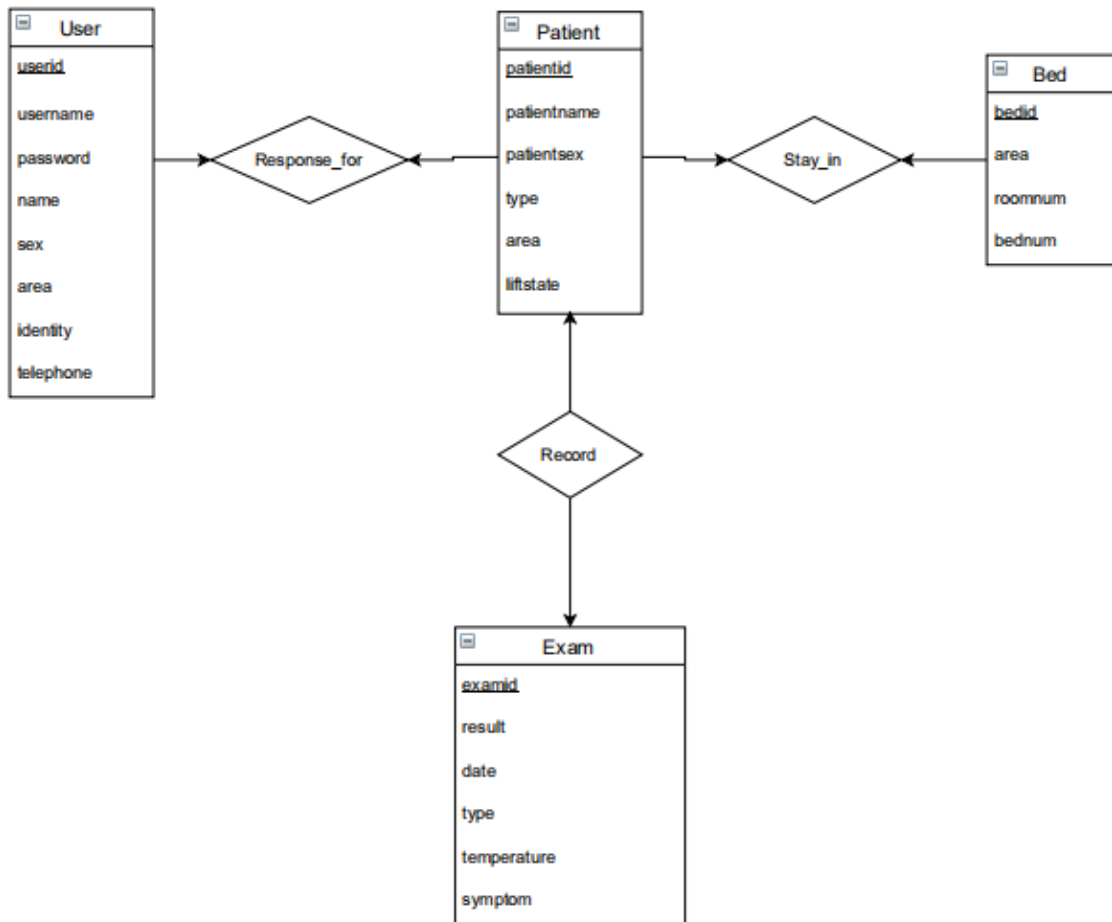
[触发器](#)

## 数据库设计ER图

基于本次要求实现的功能，我们设计了以下的ER图，以病人为中心，有负责照顾病患的病房护士，有与之对应的病房床位，还有每天由主治医生或者是病房护士为其增加的核酸检测记录。



简化版的ER图：



## 表结构说明

基于以上的ER图设计，转到关系数据库中一共设计了`user`,`patient`,`exam`,`news`四个表结构，创建语句如下：

```

create table user(
    userid int AUTO_INCREMENT,
    username varchar(20) not null,
    password varchar(20) not null,
    name varchar(20),
    sex enum('male','female'),
    area int /*0 for 轻症区,1 for 重症区,2 for 危重症区*/
    identity int /*0 for 主治医生,1 for 护士长,2 for 急诊护士,3 for 病房护士*/
    telephone varchar(20),
    patientsid varchar(255),
    primary key(userid)
)
  
```

```
create table patient(  
    patientid int AUTO_INCREMENT,  
    userid int,  
    bedid int,  
    patientname varchar(10) not null,  
    patientsex enum('male','female'),  
    type int /*0 for 轻症,1 for 重症,2 for 危重症*/  
    area int /*0 for 轻症区,1 for 重症区,2 for 危重症区,3 for 隔离区,4 for 已出院*/  
    lifestate /*0 for 已康复,1 for 住院治疗,2 for 已病亡*/  
    primary key(patientid),  
    foreign key(bedid) references bed(bedid),  
    foreign key(userid) references user(userid)  
)
```

```
create table bed(  
    bedid int AUTO_INCREMENT,  
    patientid int,  
    area int,  
    roomnum int,  
    bednum int,  
    primary key(bedid),  
    foreign key(patientid) references patient(patientid)  
)
```

```
create table exam(  
    examid int AUTO_INCREMENT,  
    patientid int,  
    result enum('positive','negative'),  
    date date,  
    type int,  
    temperature double precision,  
    symptom varchar(50),  
    primary key(examid),  
    foreign key(patientid) references patient(patientid)  
)
```

```
create table news(  
    newsid int AUTO_INCREMENT,  
    patientid int,  
    patientname varchar(10) not null,  
    originalarea int,  
    area int,  
    primary key(newsid),  
    foreign key(patientid) references patient(patientid)  
)
```

## 索引定义说明

在user表上的索引:

```
create index userid_index on user(userid);  
create index username_index on user(name);
```

在patient表上的索引：

```
create index patientid_index on patient(patientid);
create index area_index on patient(area);
```

在bed表上的索引：

```
create index bedid_index on bed(bedid);
create index bed_area_index on bed(area);
```

在exam表上的索引：

```
create index exam_patientid_index on exam(patientid);
```

在news表上的索引：

```
create index news_area_index on news(area);
create index news_id on news(newsid);
```

## 核心功能的SQL语句

本次lab中有很多对数据库的五个表的增删改查，这里分别举出一些比较核心的SQL语句。

急诊护士、主治医生、护士长和病房护士都需要查看自己负责的或者所有的病人，这对应于下面的语句：

```
select *
from patient
where area = 0
```

并且主治医生还需要看到这些病人对应的负责病房护士和床位：

```
select patientname, name, bednum, roomnum
from user natural join patient natural join bed
where area = 0
```

主治医生也需要查看自己区域内的所有病房护士：

```
select *
from user
where area = 0 and type = 3
```

主治医生能够更新自己负责的病人的状态：

```
update patient
set type = new_type, lifestate = new_lifestate
where patient_id = id_instance
```

主治医生和病房护士都可以为自己的病人增加核酸检测记录：

```
insert into exam(patientid,result,date,type,temperature,symptom)
values(id_instance,'positive',0,38.0,'fever')
```

护士长能够对病房护士进行增加或者删除：

```
insert into user(username,password,name,sex,area,identity,telephone,patientsid)
values('nurse','password','xiaohong','male',0,3,'1111111111',null);
delete from user
where userid = 89;
```

由于本次lab涉及到很多零散的与数据库的交互，但功能基本与上述类似，故不再赘述。

## 触发器

---

本次lab中应该有级联转入，病人死亡等触发器，例如一个病人死亡后，与之关联的病房护士和病床的关系应该接触；当一个病人转移区域后，可能会有别的病人重新转入他原来所在的区域。由于这些功能较为复杂，而且需要判断条件的检验，故我使用高级语言来处理这些“触发器”，而不是直接通过sql的形式设置。