**图书馆座位预约管理系统**

|  |  |
| --- | --- |
| **姓 名** | 孙友田 |
| **班 级** | 17信管1班 |
| **学 号** | 117060300104 |

目录

[1 需求分析](#_Toc10477_WPSOffice_Level1) [1](#_Toc10477_WPSOffice_Level1)

[1.1 编写目的](#_Toc3634_WPSOffice_Level2) [1](#_Toc3634_WPSOffice_Level2)

[1.2 背景](#_Toc25232_WPSOffice_Level2) [1](#_Toc25232_WPSOffice_Level2)

[1.3 定义](#_Toc3067_WPSOffice_Level2) [1](#_Toc3067_WPSOffice_Level2)

[1.4 目标](#_Toc22039_WPSOffice_Level2) [1](#_Toc22039_WPSOffice_Level2)

[1.5 需求分析](#_Toc14039_WPSOffice_Level2) [2](#_Toc14039_WPSOffice_Level2)

[1.5.1 系统总体需求](#_Toc3634_WPSOffice_Level3) [2](#_Toc3634_WPSOffice_Level3)

[1.5.2 软硬件平台设计](#_Toc25232_WPSOffice_Level3) [4](#_Toc25232_WPSOffice_Level3)

[2 概念设计](#_Toc3634_WPSOffice_Level1) [5](#_Toc3634_WPSOffice_Level1)

[2.1 实体座位与学习区域之间的联系](#_Toc24782_WPSOffice_Level2) [5](#_Toc24782_WPSOffice_Level2)

[2.2 座位和学生之间的联系](#_Toc8697_WPSOffice_Level2) [6](#_Toc8697_WPSOffice_Level2)

[2.3学生和黑名单之间的联系](#_Toc30263_WPSOffice_Level2) [6](#_Toc30263_WPSOffice_Level2)

[2.4 总E-R图](#_Toc29553_WPSOffice_Level2) [7](#_Toc29553_WPSOffice_Level2)

[3 逻辑设计](#_Toc25232_WPSOffice_Level1) [8](#_Toc25232_WPSOffice_Level1)

[4 物理设计](#_Toc3067_WPSOffice_Level1) [12](#_Toc3067_WPSOffice_Level1)

[5 触发器](#_Toc22039_WPSOffice_Level1) [13](#_Toc22039_WPSOffice_Level1)

[6 存储过程](#_Toc14039_WPSOffice_Level1) [14](#_Toc14039_WPSOffice_Level1)

[7 数据库脚本](#_Toc24782_WPSOffice_Level1) [16](#_Toc24782_WPSOffice_Level1)

[8 系统登录页面](#_Toc8697_WPSOffice_Level1) [21](#_Toc8697_WPSOffice_Level1)

[9 Django后台管理](#_Toc30263_WPSOffice_Level1) [21](#_Toc30263_WPSOffice_Level1)

### 1 需求分析

#### 1.1 编写目的

使用该文档的使用成员为实现图书馆座位预约管理系统的开发人员以及学校的图书馆具体组织管理人员。

#### 1.2 背景

项目名称：图书馆座位预约管理系统

项目提出者：孙友田

项目开发者：孙友田

项目鉴定者：刘伟洁老师

项目开始时间：2019年10月8日——2019年10月12日

#### 1.3 定义

系统数据流程图的一些定义



#### 1.4 目标

图书馆座位预约管理系统是为了解决校园图书馆座位预约管理不方便的问题，提高图书馆座位的使用效率，更加方便图书馆管理员的管理和学生的使用，所以开发该系统来解决这一问题。

图书馆座位预约管理系统用户分为两类：

学生用户：通过网址对图书馆座位预约管理系统进行访问，从而预约选座。学生可以在手机、平板、电脑等联网设备上进行座位的预约，方便快捷。

管理员用户：具备计算机操作的基础知识，能够熟练运用windows操作系统。

#### 1.5 需求分析

##### 1.5.1 系统总体需求

**(l) 通过对图书馆各个步骤、所需的各项信息等的分析，规定：**

①账户：系统技术人员设定默认的管理员账户（账号和密码）；

默认图书馆管理员已经知道自己的账号和密码；

默认学生已经知道自己的账号（即学号）．

②权限：管理员可以修改学生信息（如：密码找回），管理座位（座位的增加、减少）， 管理学习区域（开放状态），管理预约历史和黑名单（增、删、查、改）；但 是管理员没有权限查看和修改包括自己在内任何管理员的账户信息；

学生学生可以修改除自己账号外其他的个人信息（如：密码等）；也可以查 看自己的预约历史和黑名单，但是没有权限修改；同时也没有权限在自己的账 户中查看到其用户的任何信息．

③图书馆学习区域开放时间：

上午：7:00-12:00；

下午：12:30-18:00；

晚上：18:30-21:30．

④管理员管理时间：

上午：12：00-12:30；

下午：18:00-18:30；

晚上：21:30-22:00．

管理员管理操作：通过计算机对学习区域的开放状态进行管理；通过计算机对图书馆座位进行清理，将“使用中”或“暂留”的座位修改成“未使用”状态，并将该使用账户拉进黑名单．

注：被拉进黑名单的同学将不能进行座位预约，时长为到下一次管理员进行座位清理操作时．

⑤学生预约时间：

22:00-12:00进行图书馆座位7:00-12:00的预约；

12:30-18:00进行图书馆座位12:30-18:00的预约；

18:30-21:30进行图书馆座位18:30-21:30的预约．

注：在一个预约时段内，一个学生只能同时预约一个座位，已经被用户预约的座位将不能再次被重复预约．

**(2) 通过对图书馆各方面分析，图书馆中包含的实体有用户、学生、阅览室、座位、黑名单。包含的数据项分别如下：**

①用户账号（学/工号），密码，身份（学生，管理员）．

②学生编号（学号），姓名，性别，学院，班级，电话，状态（学习中，中途离开，离馆）．

③阅览室编号，名称，座位总数量，已使用座位的数量，状态（开放，关闭）．

④座位编号，所属阅览室编号，状态（使用中，暂留，未使用）．

⑤黑名单编号，加入时间，移出时间，学生编号，学生姓名，状态（已移出，未移出），原因．

**(3) 实体之间的联系有：**

①学生与座位之间（1：1）．

②学生与黑名单之间（1：1）．

③座位与学习区域之间（1：1）．

④管理员与座位之间（1：N）．

⑤管理员与学习区域之间（1：N）．

⑥管理员与预约历史之间（1：N）．

⑦管理员与黑名单之间（1：N）．

⑧管理员与学生之间（1：N）．

**(4)通过以上分析， 数据库包含的模块有：**

①输入模块　主要是各种信息的输入，如用户信息、学生信息、阅览室信息、座位信息、黑名单信息等．

②输出模块　主要是各种查询结果的输出等．

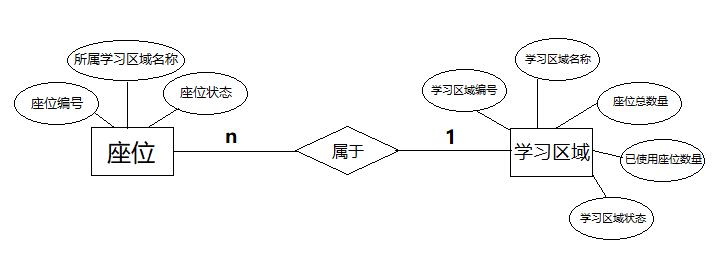
##### 1.5.2 ****软硬件平台设计****

**(1) 软件平台： windows10操作系统、oracle数据库、JetBrains PyCharm 2018.3.5 x64系统设计**

**(2) 硬件平台：普通微机**

### 2 概念设计

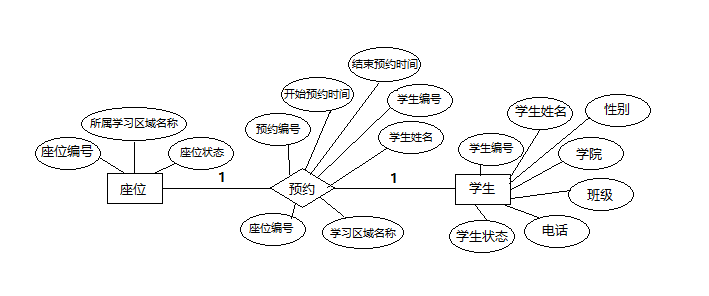
#### 2.1 实体座位与学习区域之间的联系



**相关说明：**

上述E-R图表示一个学习区域有多个座位，一个座位只能属于一个学习区域。

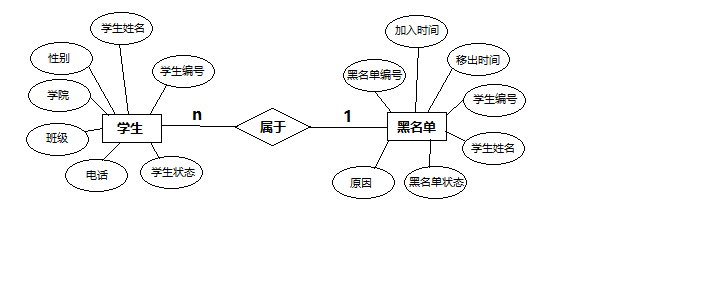
#### 2.2 座位和学生之间的联系



**相关说明：**

上述E-R图表示在一个预约时段内，一个学生只能预约一个座位，一个座位也只能被一个学生预约。

#### 2.3学生和黑名单之间的联系

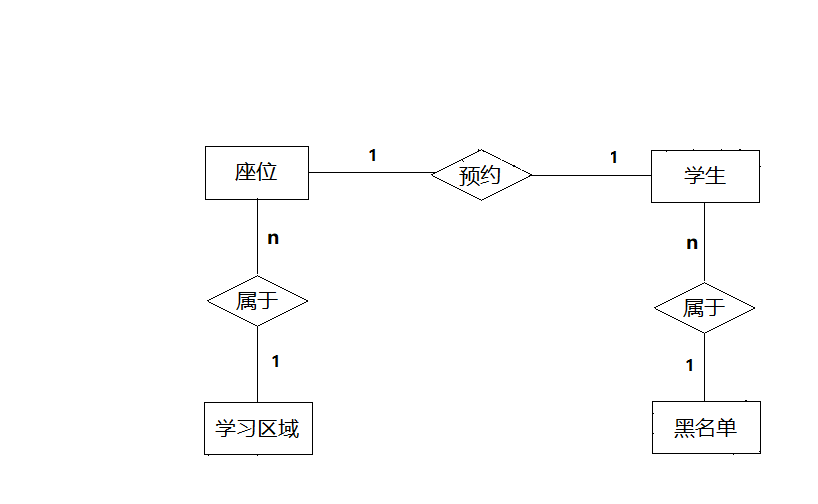


**相关说明：**

上述E-R图表示在一个工作时段内，一个学生只能被加入一次黑名单，黑名单中会包含很多个学生。

#### 2.4 总E-R图

将上述分E-R全部合并，并化简的到如下的总E-R图（为简化画图工作，上述各实体中的属性 未画出）：



上述E-R图中座位与学习区域之间的联系如上述1中的E-R图一样；座位和学生之间表示的联系如上述2中的E-R图一样；学生与黑名单之间表示的联系如上述3中的E-R图一样。

### 3 逻辑设计

本次设计的数据库在**oracle**上实现，将概念结构设计中的E-R图转换成**oracle**支持的关系数据模型后，结合上述分析得到如下数据表：

（1）用户信息表（User）：主键为账号user\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| user\_id | char(20) | not null | primary key | 账号 |
| user\_passwd | char(20) | not null |  | 密码 |
| user\_identity | char(20) | not null |  | 身份 |

(2) 学生信息表（Student）：主键为学生编号student\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| student\_id | char(20) | not null | primary key | 学生编号 |
| student\_name | char(20) | not null |  | 学生姓名 |
| student\_sex | char(20) | not null |  | 性别 |
| student\_institute | char(20) | not null |  | 学院 |
| student\_class | char(20) | not null |  | 班级 |
| student\_phone | char(20) | not null |  | 电话 |
| student\_state | char(20) | not null |  | 学生状态 |

(3) 学习区域信息表(Readroom)：主键为学习区域编号readroom\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| readroom\_id | char(20) | not null | Primary key | 学习区域编号 |
| readroom\_name | char(20) | not null |  | 学习区域名称 |
| readroom\_seat\_sum | char(20) | not null |  | 座位总数量 |
| readroom\_seat\_num\_now | char(20) | not null |  | 已使用座位数量 |
| readroom\_state | char(20) | not null |  | 学习区域状态 |

(4)座位信息表(Seat)：主键为座位编号seat\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| seat\_id | char(20) | not null | Primary key | 座位编号 |
| readroom\_name | char(20) | not null |  | 所属学习区域名称 |
| seat\_state | char(20) | not null |  | 座位状态 |

(5) 黑名单信息表(Blacklist)：主码为黑名单编号blacklist\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| blacklist\_id | char(20) | not null | Primary key | 黑名单编号 |
| add\_time | char(20) | not null |  | 加入时间 |
| end\_time | char(20) | not null |  | 移出时间 |
| student\_id | char(20) | not null | Foreign key | 学生编号 |
| student\_name | char(20) | not null |  | 学生姓名 |
| blacklist\_state | char(20) | not null |  | 黑名单状态 |
| reason | char(20) | not null |  | 原因 |

(6)预约历史信息表(Book)：主键是预约历史编号book\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| book\_id | char(20) | not null | Primary key | 预约历史编号 |
| book\_start\_time | char(20) | not null |  | 开始预约时间 |
| book\_finish\_time | char(20) | not null |  | 结束预约时间 |
| student\_id | char(20) | not null | Foreign key | 学生编号 |
| student\_name | char(20) | not null |  | 学生姓名 |
| readroom\_name | char(20) | not null |  | 学习区域名称 |
| seat\_id | char(20) | not null | Foreign key | 座位编号 |

### 4 物理设计

为了加速表的查询，根据oracle数据库对表设计的特点，对用户信息表中的‘账号user\_id’，学生信息表中的‘学生编号student\_id’，学习区域信息表中的‘学习区域编号readroom\_id’，座位信息表中的‘座位编号seat\_id’，黑名单信息表中的‘黑名单编号blacklist\_id’和预约历史信息表中的‘预约历史编book\_id’分别建立索引.

### 5 触发器

##### 建立触发器T1（当一个座位被预约成功时，座位表中的座位状态也随之更改）：

create or replace trigger tri\_1

after insert on Book

for each row

begin

update Seat set seat\_state='使用中' where seat\_id=:new.seat\_id;

end;

/

##### 建立触发器T2（当一个座位被清理时，座位表中的座位状态也随之更改）：

create or replace trigger tri\_2

after delete on Book

for each row

begin

update Seat set seat\_state='未使用' where seat\_id=:old.seat\_id;

end;

/

### 6 存储过程

##### 建立存储过程pro\_1, 输入学生的学号，查询学生的预约历史信息:

create procedure pro\_1(p\_student\_id in varchar2)

as

cursor cursor\_book--定义游标

is select \* from Book where student\_id = p\_student\_id;

cur\_record book%rowtype;--格式设置

begin

open cursor\_book;

loop

fetch cursor\_book into cur\_record;

exit when cursor\_book%notfound;--判断游标中是否存在数据

dbms\_output.put\_line('预约编号：'||cur\_record.book\_id);

dbms\_output.put\_line('开始预约时间：'||cur\_record.book\_start\_time);

dbms\_output.put\_line('结束预约时间：'||cur\_record.book\_finish\_time);

dbms\_output.put\_line('学生编号：'||cur\_record.student\_id);

dbms\_output.put\_line('学生姓名：'||cur\_record.student\_name);

dbms\_output.put\_line('阅览室名称：'||cur\_record.readroom\_name);

dbms\_output.put\_line('座位编号：'||cur\_record.seat\_id);

end loop;

close cursor\_book;

end;

/

--调用存储过程

exec pro\_1('201903');

##### 建立存储过程pro\_2, 输入学生的学号，查询学生的黑名单信息

create procedure pro\_2(p\_student\_id in varchar2)

as

cursor cursor\_blacklist--定义游标

is select \* from Blacklist where student\_id = p\_student\_id;

cur\_record Blacklist%rowtype;--格式设置

begin

open cursor\_blacklist;

loop

fetch cursor\_blacklist into cur\_record;

exit when cursor\_blacklist%notfound;--判断游标中是否存在数据

dbms\_output.put\_line('黑名单编号：'||cur\_record.blacklist\_id);

dbms\_output.put\_line('加入时间：'||cur\_record.add\_time);

dbms\_output.put\_line('结束时间：'||cur\_record.end\_time);

dbms\_output.put\_line('学生编号：'||cur\_record.student\_id);

dbms\_output.put\_line('学生姓名：'||cur\_record.student\_name);

dbms\_output.put\_line('黑名单状态：'||cur\_record.blacklist\_state);

dbms\_output.put\_line('原因：'||cur\_record.reason);

end loop;

close cursor\_blacklist;

end;

/

--调用存储过程

exec pro\_2('201902');

### 7 数据库脚本

--创建表

CREATE table User\_info--用户信息表

(

user\_id varchar2(20),--账号

user\_passwd varchar2(20),--密码

user\_identity varchar2(20)--身份（学生，管理员）

);

CREATE table Student--学生信息表

(

student\_id varchar2(20),

student\_name varchar2(20),

student\_sex varchar2(20),

student\_institute varchar2(20),--学院

student\_class varchar2(20),

student\_phone varchar2(20),

student\_state varchar2(20)--状态（学习中，中途离开，离馆）

);

CREATE table Readroom--学习区域信息表

(

readroom\_id varchar2(20),

readroom\_name varchar2(20),--名称

readroom\_seat\_sum varchar2(20),--座位总数量

readroom\_seat\_num\_now varchar2(20),--已使用座位数量

readroom\_state varchar2(20)--状态（开放，关闭）

);

CREATE table Seat--座位信息表

(

seat\_id varchar2(20),

readroom\_name varchar2(20),--所属学习区域名称

seat\_state varchar2(20)--状态（使用中，未使用）

);

CREATE table Blacklist--黑名单信息表

(

blacklist\_id varchar2(20),

add\_time varchar2(20),--加入时间

end\_time varchar2(20),--移出时间

student\_id varchar2(20),

student\_name varchar2(20),

blacklist\_state varchar2(20),--状态（已移出，未移出）

reason varchar2(20)--原因

);

CREATE table Book--预约历史信息表（映射关系产生的表，不是实体）

(

book\_id varchar2(20),

book\_start\_time varchar2(20),--开始预约时间

book\_finish\_time varchar2(20),--结束预约时间

student\_id varchar2(20),

student\_name varchar2(20),

readroom\_name varchar2(20),

seat\_id varchar2(20)

);

--添加主键约束

alter table User\_info

add CONSTRAINT pk\_user\_id PRIMARY KEY(user\_id);

alter table Student

add CONSTRAINT pk\_student\_id PRIMARY KEY(student\_id);

alter table Readroom

add CONSTRAINT pk\_readroom\_id PRIMARY KEY(readroom\_id);

alter table Seat

add CONSTRAINT pk\_seat\_id PRIMARY KEY(seat\_id);

alter table Blacklist

add CONSTRAINT pk\_blacklist\_id PRIMARY KEY(blacklist\_id);

alter table Book

add CONSTRAINT pk\_book\_id PRIMARY KEY(book\_id);

--添加外键约束

alter table Blacklist

add constraint fk\_student\_id foreign key(student\_id )

references Student(student\_id);

alter table Book

add constraint fk\_student\_id2 foreign key(student\_id )

references Student(student\_id);

alter table Book

add constraint fk\_seat\_id foreign key(seat\_id)

references Seat(seat\_id);

--输入数据

--向用户信息表添加的数据

insert into User\_info values('0001','0001','管理员');

insert into User\_info values('201901','201901','学生');

insert into User\_info values('201902','201902','学生');

insert into User\_info values('201903','201903','学生');

--向学生信息表添加的数据

insert into Student values('201901','孙友田','女','信息工程学院','17信管1班','12345678910','离馆');

insert into Student values('201902','张晓明','男','外国语学院','18翻译2班','12345678911','离馆');

insert into Student values('201903','魏小白','女','陶瓷艺术学院','18雕塑1班','12345678912','学习中');

--向学习区域信息表添加的数据

insert into Readroom values('01','负一楼自习室','150','1','开放');

--向座位信息表添加的数据

insert into Seat values('001','负一楼自习室','使用中');

insert into Seat values('002','负一楼自习室','未使用');

--向黑名单信息表添加的数据

insert into Blacklist values('1','2019-10-11 12:00','2019-10-11 18：00','201902','张晓明','未移出','未及时归还');

--向预约历史信息表中添加信息

insert into Book values('1','2019-10-11 8:00','2019-10-11 8：05','201903','魏小白','负一楼自习室','001');

insert into Book values('2','2019-10-11 8:00','2019-10-11 8：05','201901','孙友田','负一楼自习室','002');

### 8 系统登录页面



### 9 Django后台管理

