项目编号：

文档版本： 第 1 版

分册名称： 第 1册/共 1册

基于自动驾驶图片去噪的图片社区数据库设计文档

指导老师：刘琴

开发小组：君士坦丁堡队

湖南大学信息科学与工程学院软件工程2021级4班

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 总页数 |  | 正文 |  | 附录 | 1 | 生效日期： |
| 编制： | | | 审核： | | | 批准： |

文档信息

|  |  |
| --- | --- |
| 文档名称 | 基于自动驾驶图片去噪的图片社区系统项目数据库设计文档 |
| 描述 | 该文档简要介绍了数据库的表、字段及其表之间的关系。 |
| 负责人 | 胡剑锋 |
| 状态 | 1.0版本 |

文档变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **时间** | **版本号** | **修改人** | **章节** | **描述** |
| 2022-10-24 | 1.0 | 胡剑锋 |  | 从设计和需求文档创建文档初稿 |

审核结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **审核人** | **审核时间** | **意见** | **签名档** | **备注** |
| 开发团队 | 2023-10-24 | 通过 |  | 数据库设计文档通过评审 |

# 1 引言

## 1.1 编写目的

本数据库设计说明书是对基于自动驾驶图片去噪的图片社区系统数据库设计的定义。包括本系统数据逻辑结构设计、数据字典以及运行环境、安全设计等。

本数据库设计说明书适合以下读者：

1. 用户
2. 系统设计人员
3. 开发人员
4. 测试人员
5. 验收人员
6. 系统维护人员

本数据库设计说明书是以下活动的依据之一：

1. 系统详细设计
2. 系统开发
3. 系统验收
4. 系统维护

## 1.2 背景

自动驾驶作为如今热门的科研方向,自动驾驶图片的去噪处理是不可或缺的一部分,通过对自动驾驶的图片进行有效且高效的去噪处理,将会使自动驾驶图像识别的准确有明显的帮助

自动驾驶所采集的图片一方面需要被系统识别,另一方面采集的图片很有可能涉及隐私,如何在保证系统功能完备的情况下保护用户的隐私安全,这就需要设计出具有可靠结构,提供安全查询,保证数据加密的数据库来存放用户的数据,同时面对自动驾驶对数据操作的时效性加上用户海量的图片信息,保证数据库的效率,包括增删改查等操作的迅速,将是设计与实现数据库的重要指标

## 1.3 术语定义

无

## 1.4 参考资料

1）《基于自动驾驶图片去噪的图片社区系统项目需求说明书》

# 2 外部设计

## 2.1 标识符和状态

数据库软件名称：MySQL

数据库名称：denoised\_picture\_community

## 2.2 使用程序

数据库使用于“基于自动驾驶图片去噪的图片社区系统”

## 2.3 命名约定

数据库命名以模块缩写加具体表的英文词汇组成，能同一规范数据库表的命名。

数据库使用MySQL开发，开发程序员、测试和分析人员需要安装MySQL并建立相同数据库或安装相关插件进行联机访问。

## 2.4 设计约定

在本系统中,数据库的设计采用PowerDesigner进行,并且采用面向对象的设计方法,首先进行对象实体的设计,最后将对象持久化到数据库中,所有的表和表之间的关联(ER图)都采用标准的PowerDesigner设计工具进行,这样能够将整个系统的设计和数据库设计有机的结合起来。

# 3 结构设计

## 3.1 概念结构设计

### 3.1.1 实体和属性定义

**3.1.1.1用户模块**

（1）

实体名称:用户账号表

属性:用户账户ID, 密码, 邮箱, 用户名, 头像ID,用户ID

约束: 主键:用户账号 外键:头像ID,用户ID unique:邮箱

(2)

实体名称:用户信息表

属性:用户ID, 注册时间,性别,电话,封禁时长

约束: 主键:用户ID

(3)

实体名称:头像表

属性:头像ID,头像图片

约束: 主键:头像ID

**3.1.1.2个人图片模块**

（1）

实体名称:原始图片表

属性:图片编号, 图片位置, 图片上传时间, 所属用户账号，图片名字，收藏标志

约束: 主键: 图片编号 外键: 用户账号(用户表)

（2）

实体名称:生成后图片表

属性:图片编号, 图片位置, 图片生成时间, 图片名字，用户账号，去噪模式, 收藏标志

约束: 主键: 图片编号 外键: 用户账号(用户表)

（3）

实体名称：图片关联表

属性: 原始图片编号, 去噪图片编号, 准确率, 清晰度

约束: 主键: {原始图片编号，去噪图片编号} 外键: 原始图片编号（原始图片表），去噪图片编号（生成后图片表）

**3.1.1.3社区图片模块**

（1）

实体名称：社区图片表

属性:图片编号, 图片位置, 图片分享时间, 分享的用户账号, 点赞数, 图片名字,图片描述

约束: 主键: 图片编号 外键: 用户账号(用户表)

**3.1.1.4管理审核模块**

（1）

实体名称:管理员表

属性:账号,密码,用户名,邮箱

约束: 主键:账号

（2）

实体名称：图片分享申请表

属性:申请图片编号,申请图片类型,图片名字,图片描述

约束: 主键: 申请图片的编号 外键: 图片编号

（3）

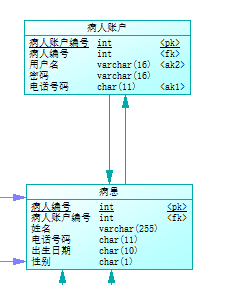
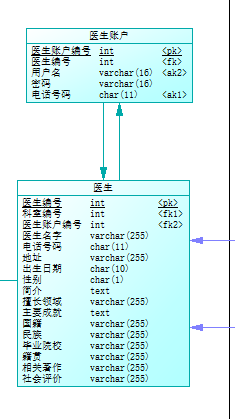
实体名称:网站每日流量表

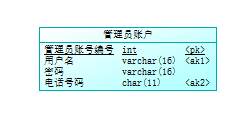
属性:日期,每日的流量,今日社区新增图片

约束: 主键: 日期

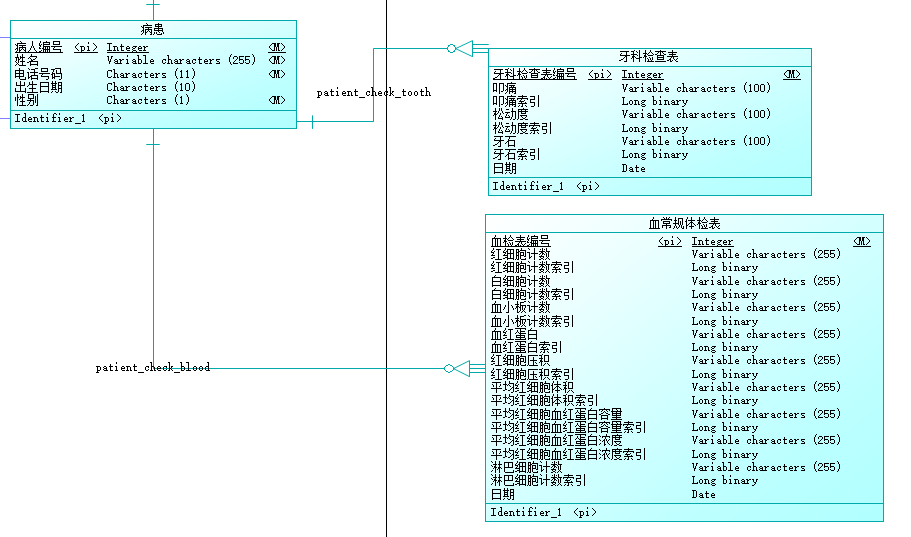
### 3.1.2 设计局部E-R模式

#### 3.1.2.1基础信息

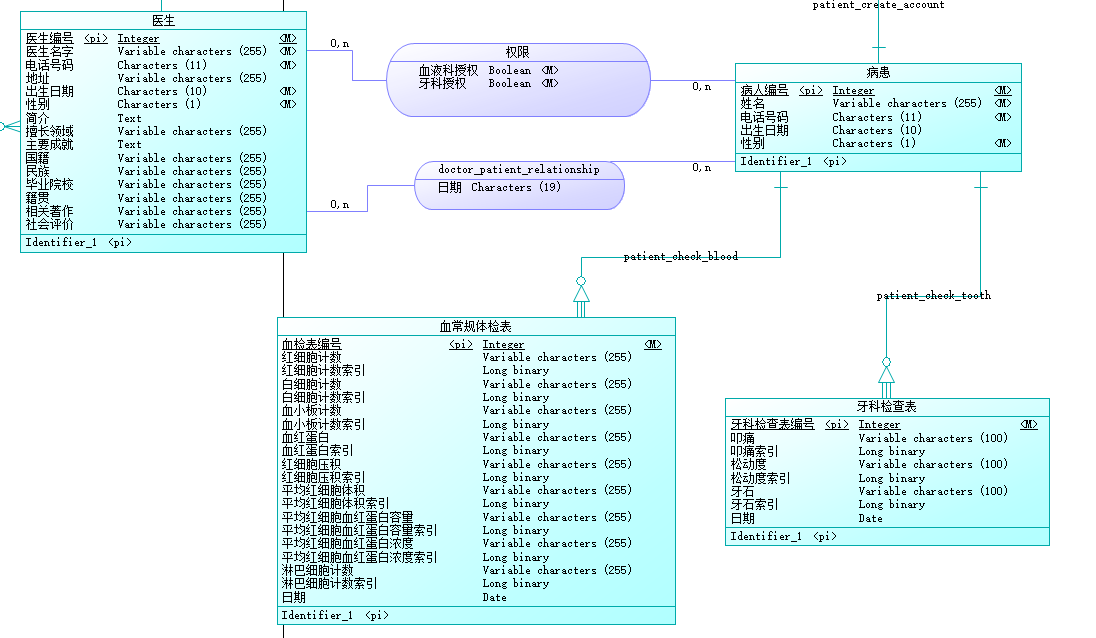




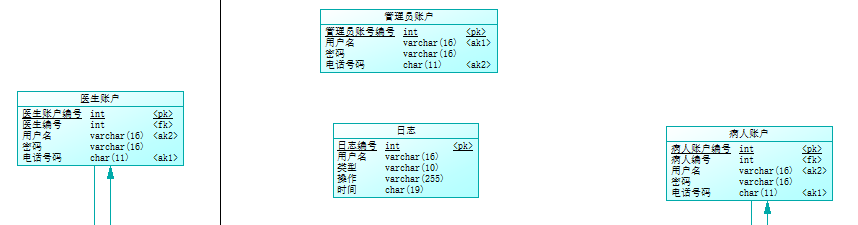
**3.1.2.2病人查看数据**



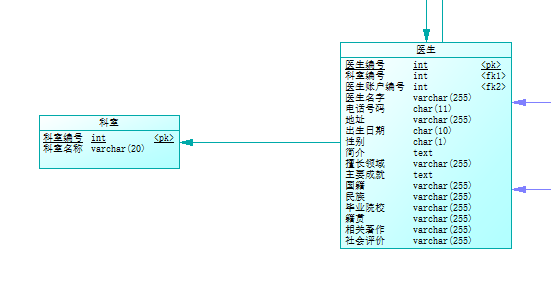
**3.1.2.3医生查看数据**



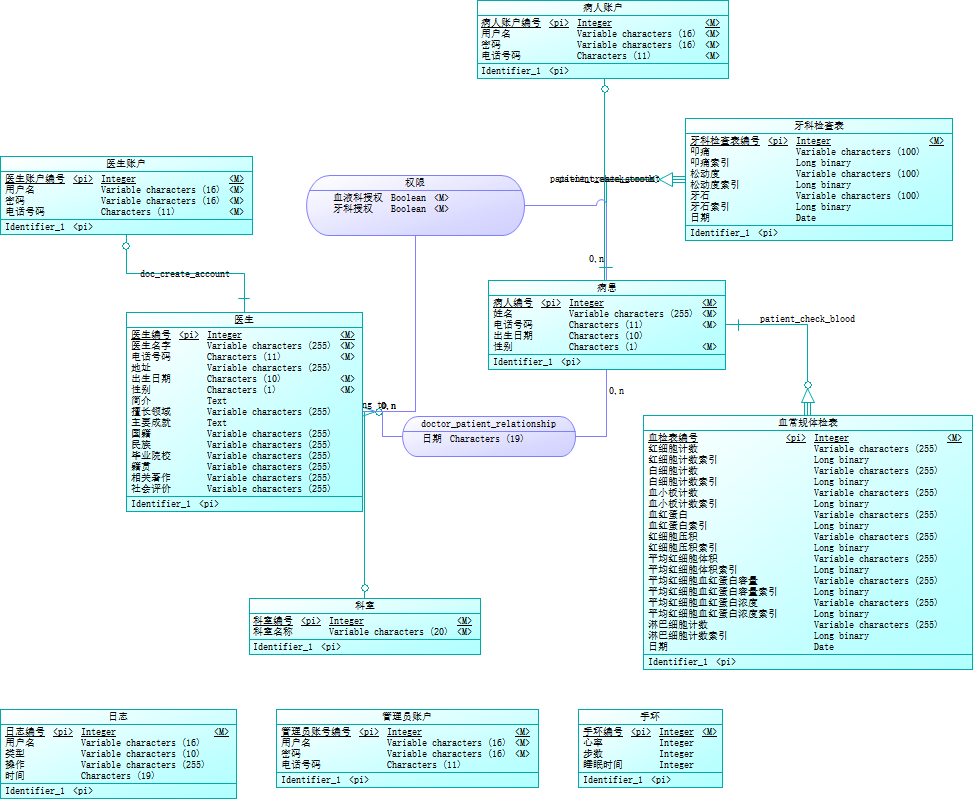
**3.1.2.4生成日志**



**3.1.2.5查看医生科室**



**3.1.2 全局E-R图**



## 3.2 逻辑结构设计

### 3.2.1汇总表

|  |  |  |
| --- | --- | --- |
| **序号** | **表名** | **功能说明** |
| 1 | t\_user\_account | 用户账号表 |
| 2 | t\_user\_information | 用户信息表 |
| 3 | t\_origin\_picture | 原始图像表 |
| 4 | t\_denoised\_picture | 去噪图像表 |
| 5 | t\_picture\_match | 图像匹配表 |
| 6 | t\_community\_picture | 社区图像表 |
| 7 | t\_admin | 管理员账户表 |
| 8 | t\_web\_information | 网站信息表 |
| 9 | t\_pending\_picture | 审核图片表 |
| 10 | t\_head\_picture | 头像表 |

### 3.2.2 表属性详细

#### 用户账号表(t\_user\_account)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 用户账户ID | user\_account\_id | int | TRUE | FALSE | #必填字段；  #自增 |
| 头像ID | head\_piciture\_id | int | FALSE | TRUE | #非空 |
| 邮箱 | email | Varchar(32) | FALSE | FALSE | #非空  #值唯一 |
| 用户密码 | user\_password | varchar(32) | FALSE | FALSE | #非空 |
| 用户名 | user\_name | varchar(15) | FALSE | FALSE | #非空 |
| 用户ID | user\_ID | Int | FALSE | TRUE | #非空 |

#### 用户信息表(t\_user\_information)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 用户ID | user\_ID | Int | FALSE | TRUE | #必填字段；  #自增； |
| 性别 | sex | char(1) | FALSE | FALSE |  |
| 手机号 | telephone | Varchar(32) | FALSE | FALSE |  |
| 注册时间 | register\_time | Datetime | FALSE | FALSE | #非空 |
| 封禁开始时间 | banned\_start\_time | Datetime | FALSE | FALSE |  |
| 封禁结束时间 | banned\_end\_time | Datetime | FALSE | FALSE |  |

#### 原始图片表(t\_origin\_picture)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 图片ID | origin\_picture\_ID | int | TRUE | FALSE | #必填字段；  #自增；  #数据从0x00000000开始自增 |
| 图片路径 | picture\_path | Varchar(255) | FALSE | FALSE | #非空 |
| 上传时间 | update\_time | Date | FALSE | FALSE | #非空 |
| 图片名字 | picture\_name | Varchar(15) | FALSE | FALSE | #非空  #名字不超过15个字符 |
| 收藏标签 | has\_collect | boolean | FALSE | FALSE | #非空  #默认为FALSE |

#### 去噪图片表(t\_denoised\_picture)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 图片ID | denoised\_picture\_ID | int | TRUE | FALSE | #必填字段；  #自增；  #数据从0x10000000开始自增 |
| 图片路径 | picture\_path | Varchar(255) | FALSE | FALSE | #非空 |
| 生成时间 | generating\_time | Date | FALSE | FALSE | #非空 |
| 图片名字 | picture\_name | Varchar(15) | FALSE | FALSE | #非空  #名字不超过15个字符 |
| 收藏标签 | has\_collect | boolean | FALSE | FALSE | #非空  #默认为FALSE |
| 去噪模式 | denoising\_mode | Varchar(20) | FALSE | FALSE | #非空 |

#### 图片匹配表(t\_picture\_match)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 原始图片ID | origin\_picture\_ID | int | TRUE | FALSE | #必填字段； |
| 去噪图片ID | denoised\_picture\_ID | int | TRUE | FALSE | #必填字段； |
| 准确度 | picture\_accuracy | double | FALSE | FALSE | #非空 |
| 清晰度 | picture\_clarity | double | FALSE | FALSE | #非空 |

#### 社区图片表(t\_comunity\_picture)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 社区图片ID | community\_picture\_ID | int | FALSE | TRUE | #必填字段；  #自增； |
| 图片路径 | picture\_path | Varchar(255) | FALSE | FALSE | #非空 |
| 生成时间 | upload\_time | Date | FALSE | FALSE | #非空 |
| 图片名字 | picture\_name | Varchar(15) | FALSE | FALSE | #非空  #名字不超过15个字符 |
| 获赞数 | picture\_favor | int | FALSE | FALSE |  |
| 图片描述 | picture\_description | Varchar(255) | FALSE | FALSE |  |

#### 管理员表(t\_admin)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 账号 | admin\_account | Varchar(40) | TRUE | FALSE | #必填字段； |
| 用户密码 | admin\_password | Varchar(32) | FALSE | FALSE | #非空 |
| 用户名 | admin\_name | Varchar(15) | FALSE | FALSE | #非空 |

#### 网站信息表(t\_web\_information)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 日期 | web\_date | Date | TRUE | FALSE | #必填字段； |
| 用户流量 | web\_user\_num | int | FALSE | FALSE | #非空 |
| 新增图片 | new\_image | int | FALSE | FALSE | #非空 |

#### 审核图片表(t\_pending\_picture)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 图片ID | pending\_picture\_ID | int | FALSE | TRUE | #必填字段；  #通过这个字段可以区分去噪前后的表 |
| 图片名字 | picture\_name | Varchar(15) | FALSE | FALSE | #非空  #名字不超过15个字符 |
| 图片描述 | picture\_description | Varchar(255) | FALSE | FALSE |  |
| 去噪状态 | denoised\_type | boolean | FALSE | FALSE | #非空 |

#### 头像表(t\_head\_picture)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***名称*** | ***代码*** | ***数据类型*** | ***主键*** | ***外来键*** | ***注释*** |
| 头像ID | head\_picture\_id | int | TRUE | FALSE | #必填字段；  #自增 |
| 头像路径 | head\_picture\_path | Varchar(255) | FALSE | FALSE | #非空 |

**3.3 物理结构设计**

数据名称：denoised\_picture\_community\_system

存储位置：默认

**3.3.1 物理设计实现**

/\*==============================================================\*/

/\* DBMS name: MySQL 5.0 \*/

/\* Created on: 2020/11/29 23:01:22 \*/

/\*==============================================================\*/

CREATE DATABASE `healthCare` CHARACTER SET 'utf8mb4';

use `healthCare` ;

drop table if exists admin\_account;

drop table if exists blood\_routine;

drop table if exists bracelet;

drop table if exists department;

drop table if exists doc\_account;

drop table if exists doctor;

drop table if exists doctor\_patient\_relationship;

drop table if exists log;

drop table if exists patient;

drop table if exists patient\_account;

drop table if exists restriction;

drop table if exists tooth\_routine;

/\*==============================================================\*/

/\* Table: admin\_account \*/

/\*==============================================================\*/

create table admin\_account

(

accountID int not null auto\_increment comment '管理员账户编号',

userName varchar(16) not null comment '用户名',

password varchar(16) not null comment '密码，长度为16位字符',

phoneNo char(11),

primary key (accountID),

unique key AK\_userName (userName),

unique key AK\_phoneNo (phoneNo)

);

alter table admin\_account comment '管理员账户';

/\*==============================================================\*/

/\* Table: blood\_routine \*/

/\*==============================================================\*/

create table blood\_routine

(

patientID int not null comment '病人的编号',

bloodNo int not null auto\_increment comment '血常规检测表编号',

rbc varchar(255) comment '正常成年人每升血液中所含红细胞的数量：3.5~5.5(单位10^12)',

rbc\_index longblob,

wbc varchar(255) comment '白细胞数量，正常成年人范围:5.0~10.0(单位10^9)',

wbc\_index longblob,

plt varchar(255) comment '血小板数量正常成年人范围:1.5~3.5(单位10^11)',

plt\_index longblob,

hb varchar(255) comment '血红蛋白数量，正常成年人范围：110-160g/L',

hb\_index longblob,

HCT varchar(255) comment '红细胞压积：参考值40-48单位(%)',

HCT\_index longblob,

MCV varchar(255) comment '平均红细胞体积，参考值80-97，单位fL',

MCV\_index longblob,

MCH varchar(255) comment '平均红细胞血红蛋白含量，参考值26.5-33.5',

MCH\_index longblob,

MCHC varchar(255) comment '平均红细胞血红蛋白浓度,参考值300-360',

MCHC\_index longblob,

LY varchar(255) comment '淋巴细胞计数，参考值0.8-4.0',

LY\_index longblob,

date date comment '检测日期:如2000-01-01 16:10:10',

primary key (bloodNo)

);

/\*==============================================================\*/

/\* Table: bracelet \*/

/\*==============================================================\*/

create table bracelet

(

braceletID int not null auto\_increment comment '手环的ID',

heartRate int comment '正常人心率60-100每分钟',

steps int comment '步数,每天走5400-7900为正常',

sleepTime int comment '睡眠时间，按分钟计算',

primary key (braceletID)

);

/\*==============================================================\*/

/\* Table: department \*/

/\*==============================================================\*/

create table department

(

departmentNo int not null auto\_increment comment '科室的编号',

departmentName varchar(20) not null comment '科室的名称，不超过20个字符',

primary key (departmentNo)

);

alter table department comment '存储科室的基本信息';

/\*==============================================================\*/

/\* Table: doc\_account \*/

/\*==============================================================\*/

create table doc\_account

(

daccountNo int not null auto\_increment comment '医生账户的NO',

doctorID int comment '医生的编号，一个编号对应一个医生',

userName varchar(16) not null comment '用户名：不超过16个字符',

password varchar(16) not null comment '密码，长度为16个字符',

phoneNo char(11) not null comment '电话号码,长度为11位字符',

primary key (daccountNo),

unique key AK\_phoneNo (phoneNo),

unique key AK\_userName (userName)

);

alter table doc\_account comment '医生的账户列表';

/\*==============================================================\*/

/\* Table: doctor \*/

/\*==============================================================\*/

create table doctor

(

doctorID int not null auto\_increment comment '医生的编号，一个编号对应一个医生',

departmentNo int comment '科室的编号',

daccountNo int comment '医生账户的NO',

doctorName varchar(255) comment '医生的名字',

phoneNo char(11) not null comment '医生的电话号码',

address varchar(255) comment '医生的家庭住址',

birthday char(10) comment '医生的出生日期,格式为：xxxx-xx-xx',

sex char(1) comment '医生的性别:''f''表示女性，''m''表示男性',

introduction text comment '该医生的简介',

expertise varchar(255) comment '该医生擅长的领域',

achievements text comment '医生成就',

nationality varchar(255) comment '医生国籍',

nation varchar(255) comment '医生民族',

college varchar(255) comment '毕业院校',

nativePlace varchar(255) comment '籍贯',

works varchar(255) comment '相关著作',

evaluation varchar(255) comment '社会评价',

primary key (doctorID)

);

alter table doctor comment '存储医生基本信息的表';

/\*==============================================================\*/

/\* Table: doctor\_patient\_relationship \*/

/\*==============================================================\*/

create table doctor\_patient\_relationship

(

patientID int not null comment '病人的编号',

doctorID int not null comment '医生的编号，一个编号对应一个医生',

date char(19),

primary key (patientID, doctorID)

);

alter table doctor\_patient\_relationship comment '该表记录了医生-病人的对应关系';

/\*==============================================================\*/

/\* Table: log \*/

/\*==============================================================\*/

create table log

(

logNo int not null auto\_increment comment '日志编号',

userName varchar(16) comment '执行动作的用户名',

type varchar(10),

action varchar(255) comment '记录用户所做的操作，不能超过255个字符',

time char(19) comment '记录操作的时间点：格式为xxxx-xx-xx xx:xx:xx',

primary key (logNo)

);

alter table log comment '日志信息：存储某一个用户在某一个时间点做了什么操作';

/\*==============================================================\*/

/\* Table: patient \*/

/\*==============================================================\*/

create table patient

(

patientID int not null auto\_increment comment '病人的编号',

paccountNo int comment '病人的账户编号',

patientName varchar(255) comment '病人的名字',

phoneNo char(11) not null comment '病人的手机号码',

birthday char(10) comment '病人的生日',

sex char(1) comment '病人的性别：f表示女性，m表示男性',

primary key (patientID)

);

alter table patient comment '存储病人的基本信息';

/\*==============================================================\*/

/\* Table: patient\_account \*/

/\*==============================================================\*/

create table patient\_account

(

paccountNo int not null auto\_increment comment '病人的账户编号',

patientID int comment '病人的编号',

userName varchar(16) not null comment '用户名，最多16个字符',

password varchar(16) not null comment '密码，长度为16个字符',

phoneNo char(11) not null comment '电话号码',

primary key (paccountNo),

unique key AK\_phoneNo (phoneNo),

unique key AK\_userName (userName)

);

alter table patient\_account comment '病人的账户信息';

/\*==============================================================\*/

/\* Table: restriction \*/

/\*==============================================================\*/

create table restriction

(

patientID int not null comment '病人的编号',

doctorID int not null comment '医生的编号，一个编号对应一个医生',

blood\_grant bool not null default 1,

tooth\_grant bool not null default 1,

primary key (patientID, doctorID)

);

/\*==============================================================\*/

/\* Table: tooth\_routine \*/

/\*==============================================================\*/

create table tooth\_routine

(

patientID int not null comment '病人的编号',

toothNo int not null auto\_increment comment '表的编号',

pain varchar(100) comment '牙科检查之叩痛数据项：由轻到重分五个等级，用字符1-5表示',

pain\_index longblob,

mobility varchar(100) comment '松动度检测，不松动-严重松动由1-3表示',

mobility\_index longblob,

tartar varchar(100) comment '有无牙石，牙石数量从少到多用1-5表示',

tartar\_index longblob,

date date comment '检测日期：年-月-日 时:分:秒共19位字符',

primary key (toothNo)

);

alter table tooth\_routine comment '病人的牙科检查表';

alter table blood\_routine add constraint FK\_patient\_check\_blood foreign key (patientID)

references patient (patientID) on delete restrict on update restrict;

alter table doc\_account add constraint FK\_doc\_create\_account foreign key (doctorID)

references doctor (doctorID) on delete restrict on update restrict;

alter table doctor add constraint `FK\_belong to` foreign key (departmentNo)

references department (departmentNo) on delete restrict on update restrict;

alter table doctor add constraint FK\_doc\_create\_account foreign key (daccountNo)

references doc\_account (daccountNo) on delete restrict on update restrict;

alter table doctor\_patient\_relationship add constraint FK\_doctor\_patient\_relationship foreign key (doctorID)

references doctor (doctorID) on delete restrict on update restrict;

alter table doctor\_patient\_relationship add constraint FK\_doctor\_patient\_relationship foreign key (patientID)

references patient (patientID) on delete restrict on update restrict;

alter table patient add constraint FK\_patient\_create\_account foreign key (paccountNo)

references patient\_account (paccountNo) on delete restrict on update restrict;

alter table patient\_account add constraint FK\_patient\_create\_account foreign key (patientID)

references patient (patientID) on delete restrict on update restrict;

alter table restriction add constraint FK\_restriction foreign key (doctorID)

references doctor (doctorID) on delete restrict on update restrict;

alter table restriction add constraint FK\_restriction foreign key (patientID)

references patient (patientID) on delete restrict on update restrict;

alter table tooth\_routine add constraint FK\_patient\_check\_tooth foreign key (patientID)

references patient (patientID) on delete restrict on update restrict;

**4 运用设计**

**4.1 数据字典设计**



**4.2 安全设计**

智慧医疗系统数据库安装在指定的服务器上。用户对数据库进行操作时，不对数据库直接访问，而是通过登录WEB平台软件和APP对数据库进行访问，这样避免了用户直接操作数据库过程中可能产生的不当操作。