项目编号：

文档版本： 第 1 版

分册名称： 第 1册/共 1册

人体生理数据监测系统

数据库设计文档

指导老师：荣辉桂

开发小组：代码怎么敲都队

湖南大学信息科学与工程学院软件工程2020级4班

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 总页数 | 27 | 正 文 | 23 | 附 录 | 1 | 生效日期：2022-10-25 |
| 编制：崔光博、海日娜、安冠东、刘文韬、冯秋怡 | | | 审核： 荣辉桂 | | | 批准：边耐政 |

文档信息

|  |  |
| --- | --- |
| **文档名称** | 人体生理数据监测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** | 刘文韬 |
| **状态** | 1.0版本 |

文档变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **时间** | **版本号** | **修改人** | **章节** | **描述** |
| 2022-10-25 | 初稿 | 海日娜，刘文韬，崔光博，冯秋怡，安冠东 | 所有章节 | 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 | 2022-10-26 | 通过 |  | 数据库设计文档通过评审 |

目录

[1 引言 5](#_Toc118055693)

[1.1 编写目的 5](#_Toc118055694)

[1.2 背景 5](#_Toc118055695)

[1.3 术语定义 6](#_Toc118055696)

[1.4 参考资料 6](#_Toc118055697)

[2 外部设计 6](#_Toc118055698)

[2.1 标识符和状态 6](#_Toc118055699)

[2.2 使用程序 6](#_Toc118055700)

[2.3 命名约定 6](#_Toc118055701)

[2.4 设计约定 6](#_Toc118055702)

[3 结构设计 7](#_Toc118055703)

[3.1 概念结构设计 7](#_Toc118055704)

[3.1.1 实体和属性定义 7](#_Toc118055705)

[3.1.1.1用户模块 7](#_Toc118055706)

[3.1.1.2健康模块 8](#_Toc118055707)

[3.1.1.3饮食模块 8](#_Toc118055708)

[3.1.1.4社区模块 9](#_Toc118055709)

[3.1.2 设计局部E-R模式 10](#_Toc118055710)

[3.1.2.1 健康日志 10](#_Toc118055711)

[3.1.2.2 饮食管家 11](#_Toc118055712)

[3.1.2.3 医患交流 11](#_Toc118055713)

[3.1.2.4 木糖社区 11](#_Toc118055714)

[3.1.2.5 我的消息 12](#_Toc118055715)

[3.1.2.6 登录日志 12](#_Toc118055716)

[3.1.3 全局E-R图 12](#_Toc118055717)

[3.2 逻辑结构设计 13](#_Toc118055718)

[3.2.1汇总表 13](#_Toc118055719)

[3.2.2 表属性详细 14](#_Toc118055720)

[用户登录日志(t\_log\_login) 14](#_Toc118055721)

[用户表(t\_all\_user) 14](#_Toc118055722)

[用户个人消息(t\_personal\_message) 15](#_Toc118055723)

[管理员表(t\_background\_administrator) 15](#_Toc118055724)

[医生资格认证表(t\_community\_doctorverify) 15](#_Toc118055725)

[饮食记录表(t\_diet\_record) 16](#_Toc118055726)

[食物表(t\_communiy\_food) 16](#_Toc118055727)

[社区药品表(t\_community\_drug) 16](#_Toc118055728)

[社区关系表(t\_community\_relation) 17](#_Toc118055729)

[社区消息表(t\_community\_chat) 17](#_Toc118055730)

[社区帖子表(t\_community\_post) 17](#_Toc118055731)

[社区交互表(t\_community\_interaction) 18](#_Toc118055732)

[我的药物表(t\_health\_medication) 18](#_Toc118055733)

[健康记录表(t\_health\_record) 18](#_Toc118055734)

[健康建议表(t\_health\_suggest) 19](#_Toc118055735)

[3.3 物理结构设计 19](#_Toc118055736)

[3.3.1 物理设计实现 19](#_Toc118055737)

[4 运用设计 27](#_Toc118055738)

[4.1 数据字典设计 27](#_Toc118055739)

[4.2 安全设计 27](#_Toc118055740)

# 1 引言

## 1.1 编写目的

本数据库设计说明书是对噪声地图系统数据库设计的定义。包括本系统数据逻辑结构设计以及运行环境、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

国际糖尿病联合会在2021年出具的报告显示，糖尿病是一场前所未有的疾病大流行。在过去的10年间，我国糖尿病患者人数由9千万增加至1亿4千万。对已经发病的患者，密切监测血糖，规律就诊，通过积极有效的干预和治疗，可以尽可能地防止出现心脑血管疾病、糖尿病肾病等并发症。研究表明，糖尿病患者使用一些辅助工具管理疾病可改善自身病情，提高糖尿病患者自我监测的依从性。

本系统能够帮糖尿病患者规律进行身体健康状态监测与预警，且界面友好、知识精细的软件，从而辅助他们进行糖尿病治疗，从而守护身体健康。

## 1.3 术语定义

无。

## 1.4 参考资料

1）《人体生理数据监测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：health monitoring system

## 2.2 使用程序

数据库使用于“人体生理健康数据监测系统”

## 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

属性：用户编号、邮箱、密码、用户名、头像、身份、权限、关键词、生日、性别、血型、身高

约束：无

2）

实体名称：个人消息

属性：消息编号、用户编号、消息来源、消息内容、消息类型、消息时间

约束：外键：用户编号

3）

实体名称：登录日志表

属性：登录编号、用户编号、登录时间

约束：外键：用户编号

4）

实体名称：管理员

属性：管理员编号、密码

约束：无

#### 3.1.1.2健康模块

1）

实体名称：健康记录

属性：健康记录编号、用户编号、日期时间、血糖、高血压、低血压、血脂、心率、体重、健康状况

约束：外键：用户编号

2）

实体名称：药物信息

属性：药物编号、药物名称、药物图片、药物详情

约束：无

3）

实体名称：用药

属性：用药编号、用户编号、药物编号、用药日期、用药时间、是否用药

约束：外键：用户编号、药物编号

4）

实体名称：健康建议

属性：建议编号、健康信息类型、数值档次、健康建议

约束：无

#### 3.1.1.3饮食模块

1）

实体名称：食物信息

属性：食物编号、食物名称、描述、含糖量、蛋白质、脂肪、能量、图片

约束：无

2）

实体名称：饮食记录

属性：饮食记录编号、用户编号、日期、用餐时间、食物编号

约束：外键：用户编号、食物编号

#### 3.1.1.4社区模块

1）

实体名称：帖子

属性：帖子编号、用户编号、用户名、时间、标题、内容、图像、点赞数、收藏数

约束：外键：用户编号

2）

实体名称：互动

属性：互动编号、用户编号、帖子编号、是否点赞、是否收藏

约束：外键：用户编号、帖子编号

3）

实体名称：医生认证

属性：医生编号、用户编号、医院、身份证号、医师资格证、认证状态、管理员编号

约束：外键：用户编号、管理员编号

4）

实体名称：医患关系

属性：关系编号、医生编号、用户编号

约束：外键：医生编号、用户编号

5）

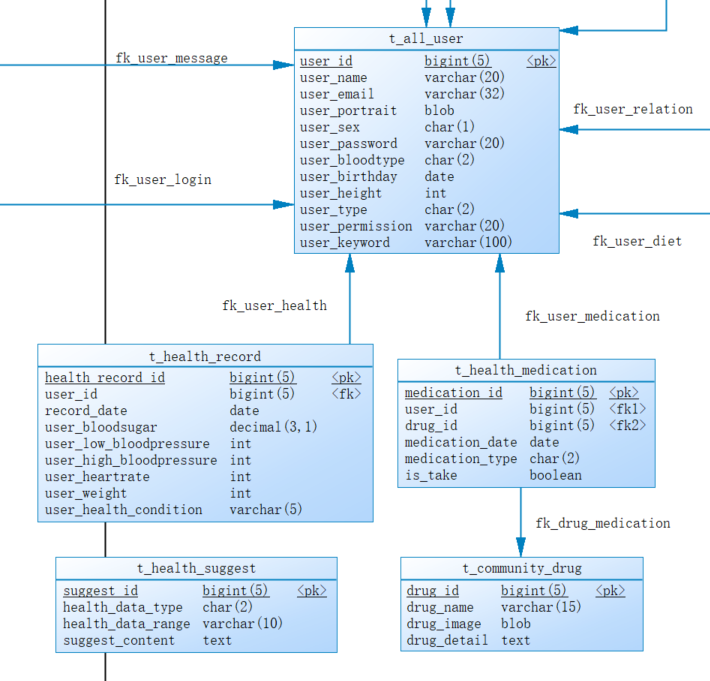
实体名称：医患聊天

属性：聊天编号、关系编号、时间、聊天内容、发出方

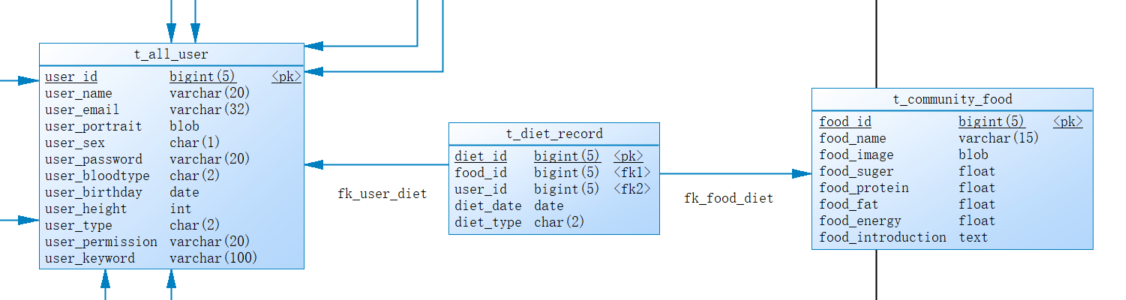
约束：外键：关系编号

### 3.1.2 设计局部E-R模式

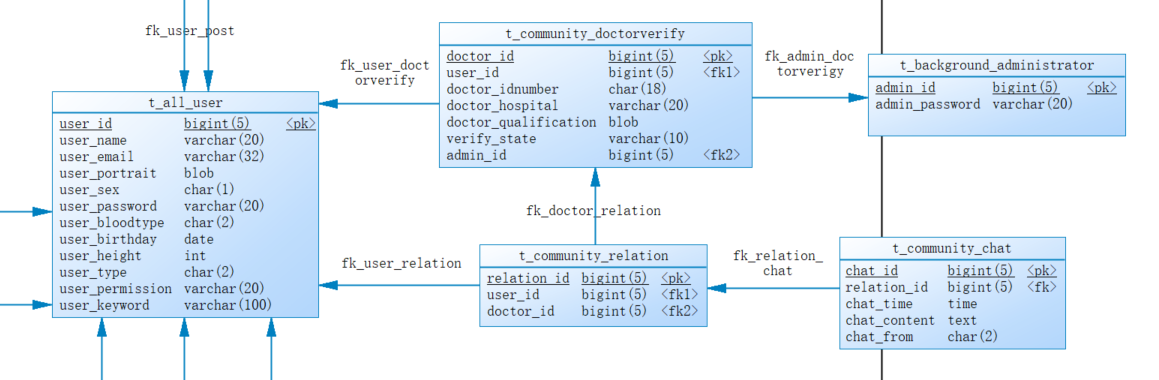
#### 3.1.2.1 健康日志



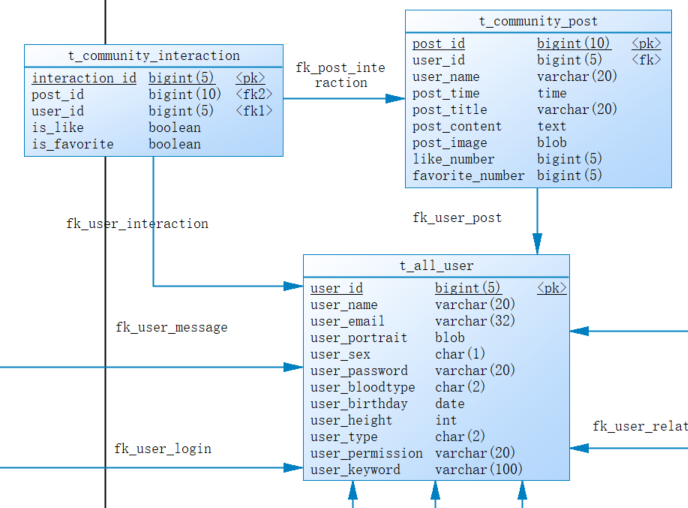
#### 3.1.2.2 饮食管家



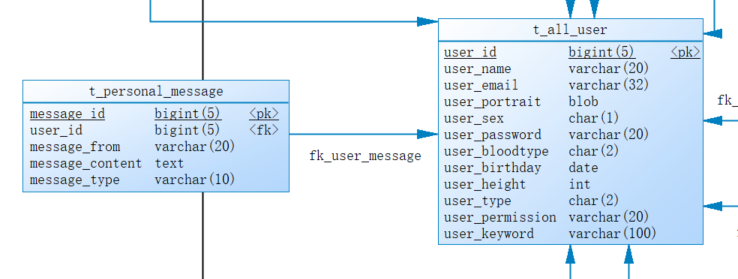
#### 3.1.2.3 医患交流



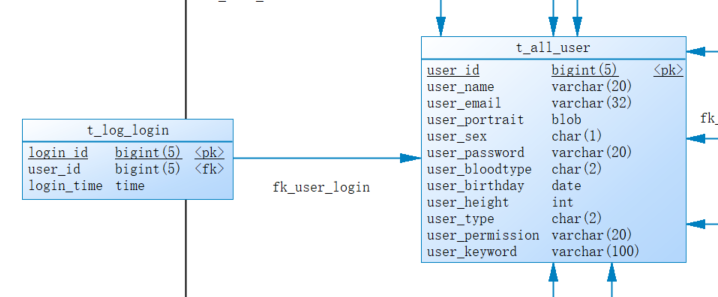
#### 3.1.2.4 木糖社区



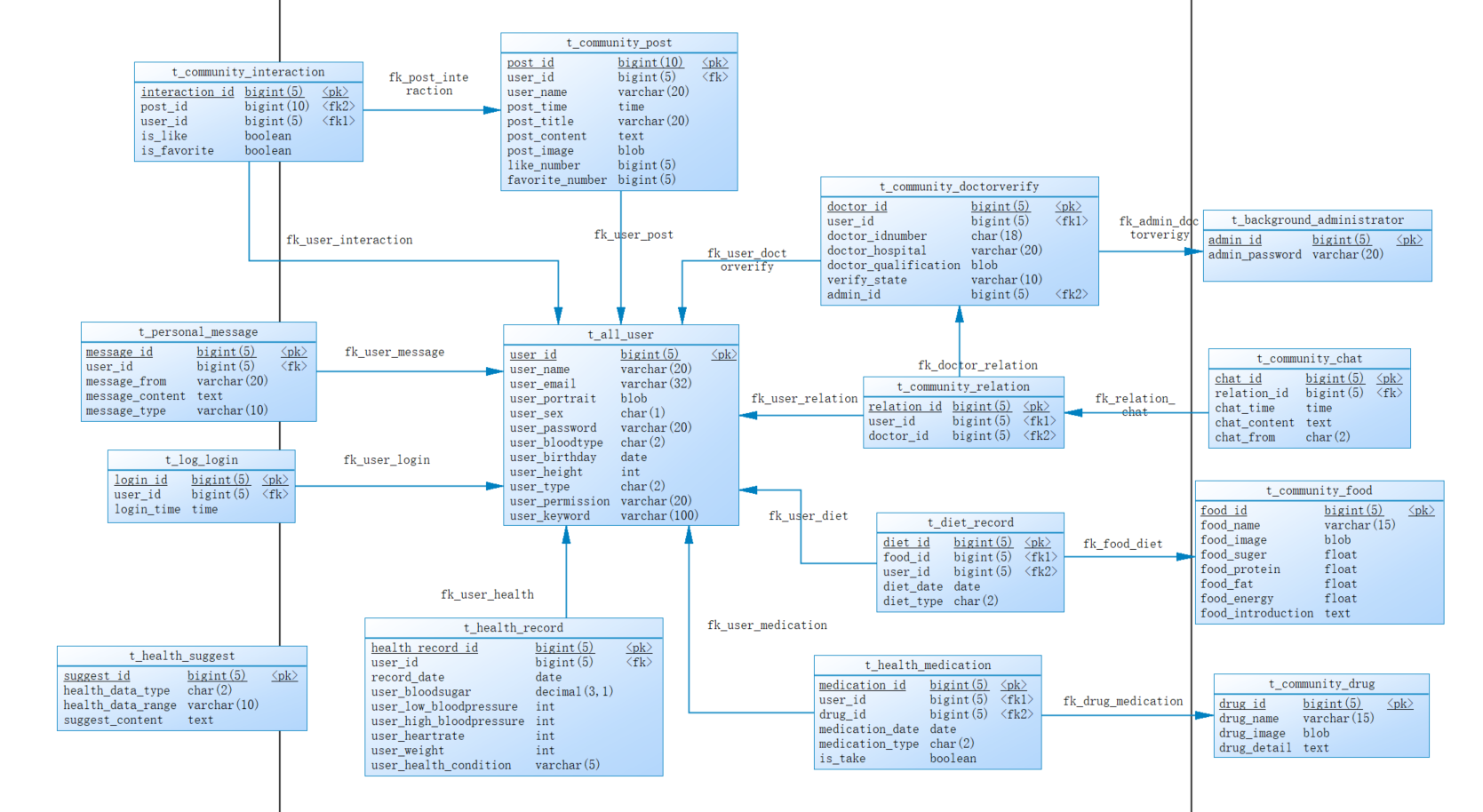
#### 3.1.2.5 我的消息



**3.1.2.6 登录日志**



### 3.1.3 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\_log\_login | 用户登录日志 |
| 2 | t\_all\_user | 用户表 |
| 3 | t\_personal\_message | 用户个人消息 |
| 4 | t\_background\_administrator | 管理员表 |
| 5 | t\_community\_doctorverify | 医生资格认证表 |
| 6 | t\_diet\_record | 饮食记录表 |
| 7 | t\_community\_food | 食物表 |
| 8 | t\_community\_drug | 社区药品表 |
| 9 | t\_community\_relation | 社区关系表 |
| 10 | t\_community\_chat | 社区消息表 |
| 11 | t\_community\_post | 社区帖子表 |
| 12 | t\_community\_interaction | 社区交互表 |
| 13 | t\_health\_medication | 我的药物表 |
| 14 | t\_health\_record | 健康记录表 |
| 15 | t\_health\_suggest | 健康建议表 |

### 3.2.2 表属性详细

#### 用户登录日志(t\_log\_login)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| login\_id | login\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| login\_time | login\_ time | time | FALSE | FALSE |  |

#### 用户表(t\_all\_user)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| user\_id | user\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_name | user\_name | varchar(20) | FALSE | FALSE |  |
| user\_email | user\_ email | varchar(32) | FALSE | FALSE |  |
| user\_portrait | user\_portrait | blob | FALSE | FALSE |  |
| user\_sex | user\_sex | char(1) | FALSE | FALSE |  |
| user\_password | user\_passord | varchar(20) | FALSE | FALSE |  |
| user\_bloodtype | user\_bloodtype | char(2) | FALSE | FLASE |  |
| user\_birthday | user\_birthday | date | FALSE | FALSE |  |
| user\_height | user\_height | int | FALSE | FALSE |  |
| user\_type | user\_type | char(2) | FALSE | FALSE |  |
| user\_permission | user\_permission | varchar(20) | FALSE | FALSE |  |
| user\_keyword | user\_keyword | varchar(100) | FALSE | FALSE |  |

#### 用户个人消息(t\_personal\_message)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| message\_id | message\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| message\_time | message\_ time | time | FALSE | FALSE |  |
| message\_from | message\_from | varchar(20) | FALSE | FALSE |  |
| message\_content | message\_content | text | FALSE | FALSE |  |
| message\_type | message\_type | varchar(10) | FALSE | FALSE |  |

#### 管理员表(t\_background\_administrator)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| admin\_id | admin\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| admin\_password | admin\_password | varchar(20) | FALSE | FALSE |  |

#### 医生资格认证表(t\_community\_doctorverify)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| doctor\_id | doctor\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| doctor\_idnumber | doctor\_idnumber | char(18) | FALSE | FALSE |  |
| doctor\_hospital | doctor\_hospital | varchar(20) | FALSE | FALSE |  |
| doctor\_qualification | doctor\_qualification | blob | FALSE | FALSE |  |
| verify\_state | verify\_state | varchar(10) | FALSE | FALSE |  |
| admin\_id | admin\_id | bigint(5) | FALSE | TRUE |  |

#### 饮食记录表(t\_diet\_record)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| diet\_id | diet\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| food\_id | food\_id | bigint(5) | FALSE | TRUE |  |
| user\_id | user\_ id | bigint(5) | FALSE | TRUE |  |
| diet\_date | diet\_date | date | FALSE | FALSE |  |
| diet\_type | diet\_type | char(2) | FALSE | FALSE |  |

#### 食物表(t\_communiy\_food)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| food\_id | food\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| food\_name | food\_name | varchar(15) | FALSE | FALSE |  |
| food\_image | food\_ image | blob | FALSE | FALSE |  |
| food\_sugar | food\_sugar | float | FALSE | FALSE |  |
| food\_protein | food\_protein | float | FALSE | FALSE |  |
| food\_fat | food\_fat | float | FALSE | FALSE |  |
| food\_energy | food\_energy | float | FALSE | FALSE |  |
| food\_introduction | food\_introduction | text | FALSE | FALSE |  |

#### 社区药品表(t\_community\_drug)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| drug\_id | drug\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| drug\_name | drug\_name | varchar(15) | FALSE | FALSE |  |
| drug\_image | drug\_image | blob | FALSE | FALSE |  |
| drug\_detail | drug\_detail | text | FALSE | FALSE |  |

#### 社区关系表(t\_community\_relation)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| relation\_id | relation\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| doctor\_id | doctor\_id | bigint(5) | FALSE | TRUE |  |

#### 社区消息表(t\_community\_chat)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| chat\_id | chat\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| relation\_id | relation\_id | bigint(5) | FALSE | TRUE |  |
| chat\_time | chat\_time | time | FALSE | FALSE |  |
| chat\_content | chat\_content | text | FALSE | FALSE |  |
| chat\_from | chat\_from | char(2) | FALSE | FALSE |  |

#### 社区帖子表(t\_community\_post)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| post\_id | post\_id | bigint(10) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| user\_name | user\_name | varchar(20) | FALSE | FALSE |  |
| post\_time | post\_time | time | FALSE | FALSE |  |
| post\_title | post\_title | varchar(20) | FALSE | FALSE |  |
| post\_content | post\_content | text | FALSE | FALSE |  |
| post\_image | post\_image | blob | FALSE | FALSE |  |
| like\_number | like\_number | bigint(5) | FALSE | FALSE |  |
| favorite\_number | favorite\_number | bigint(5) | FALSE | FALSE |  |

#### 社区交互表(t\_community\_interaction)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| interaction\_id | interaction\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| post\_id | post\_id | bigint(10) | FALSE | TRUE |  |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| is\_like | is\_like | boolean | FALSE | FALSE |  |
| is\_favorite | is\_favorite | boolean | FALSE | FALSE |  |

#### 我的药物表(t\_health\_medication)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| medication\_id | medication\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| drug\_id | drug\_id | bigint(5) | FALSE | TRUE |  |
| medication\_date | medication\_date | date | FALSE | FALSE |  |
| medication\_type | medication\_type | char(2) | FALSE | FALSE |  |
| is\_take | is\_take | boolean | FALSE | FALSE |  |

#### 健康记录表(t\_health\_record)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| health\_record\_id | health\_record\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| user\_id | user\_id | bigint(5) | FALSE | TRUE |  |
| record\_date | record\_date | date | FALSE | FALSE |  |
| user\_bloodsugar | user\_bloodsugar | decimal(3,1) | FALSE | FALSE |  |
| user\_low\_bloodpressure | user\_low\_bloodpressure | int | FALSE | FALSE |  |
| user\_high\_bloodpressure | user\_high\_bloodpressure | int | FALSE | FALSE |  |
| user\_heartrate | user\_heartrate | int | FALSE | FALSE |  |
| user\_weight | user\_weight | int | FALSE | FALSE |  |
| user\_health\_condition | user\_health\_condition | varchar(5) | FALSE | FALSE |  |

#### 健康建议表(t\_health\_suggest)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *名称* | *代码* | *数据类型* | *主要的* | *外来键* | *注释* |
| drug\_id | drug\_id | bigint(5) | TRUE | FALSE | #必填字段；模糊查询；精确查询；#自增长类型 |
| drug\_name | drug\_name | varchar(15) | FALSE | FALSE |  |
| drug\_image | drug\_image | blob | FALSE | FALSE |  |
| drug\_detail | drug\_detail | text | FALSE | FALSE |  |

## 3.3 物理结构设计

数据名称：health monitoring system

存储位置：默认

建立程序员视图，包括：

数据在内存中的安排，包括对索引区、缓冲区的设计。

所使用的外存设备及外存空间的组织，包括索引区，数据块的组织和划分。

访问数据的方式方法。

### 3.3.1 物理设计实现

/\*

Navicat Premium Data Transfer

Source Server : MySQL

Source Server Type : MySQL

Source Server Version : 80018

Source Host : localhost:3306

Source Schema : noisesystem

Target Server Type : MySQL

Target Server Version : 80018

File Encoding : 65001

\*/

SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

drop table if exists t\_all\_user;

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

/\* Table: t\_all\_user \*/

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

create table t\_all\_user

(

user\_id bigint(5) not null,

user\_name varchar(20) not null,

user\_email varchar(32) not null,

user\_portrait blob not null,

user\_sex char(1) not null,

user\_password varchar(20) not null,

user\_bloodtype char(2) not null,

user\_birthday date not null,

user\_height int not null,

user\_type char(2) not null,

user\_permission varchar(20) not null,

user\_keyword varchar(100) not null,

primary key (user\_id)

);

drop table if exists t\_background\_administrator;

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

/\* Table: t\_background\_administrator \*/

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

create table t\_background\_administrator

(

admin\_id bigint(5) not null,

admin\_password varchar(20) not null,

primary key (admin\_id)

);

drop table if exists t\_log\_login;

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

/\* Table: t\_log\_login \*/

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

create table t\_log\_login

(

login\_id bigint(5) not null,

user\_id bigint(5) not null,

login\_time time not null,

primary key (login\_id)

);

alter table t\_log\_login add constraint FK\_fk\_user\_login foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_personal\_message;

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

/\* Table: t\_personal\_message \*/

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

create table t\_personal\_message

(

message\_id bigint(5) not null,

user\_id bigint(5) not null,

message\_from varchar(20) not null,

message\_content text not null,

message\_type varchar(10) not null,

primary key (message\_id)

);

alter table t\_personal\_message add constraint FK\_pk\_user\_message foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_community\_drug;

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

/\* Table: t\_community\_drug \*/

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

create table t\_community\_drug

(

drug\_id bigint(5) not null,

drug\_name varchar(15) not null,

drug\_image blob,

drug\_detail text not null,

primary key (drug\_id)

);

drop table if exists t\_community\_food;

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

/\* Table: t\_community\_food \*/

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

create table t\_community\_food

(

food\_id bigint(5) not null,

food\_name varchar(15) not null,

food\_image blob,

food\_suger float not null,

food\_protein float not null,

food\_fat float not null,

food\_energy float not null,

food\_introduction text not null,

primary key (food\_id)

);

drop table if exists t\_community\_doctorverify;

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

/\* Table: t\_community\_doctorverify \*/

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

create table t\_community\_doctorverify

(

doctor\_id bigint(5) not null,

user\_id bigint(5) not null,

doctor\_idnumber char(18) not null,

doctor\_hospital varchar(20) not null,

doctor\_qualification blob not null,

verify\_state varchar(10) not null,

admin\_id bigint(5) not null,

primary key (doctor\_id)

);

alter table t\_community\_doctorverify add constraint FK\_fk\_admin\_doctorverigy foreign key (admin\_id)

references t\_background\_administrator (admin\_id) on delete restrict on update restrict;

alter table t\_community\_doctorverify add constraint FK\_fk\_user\_doctorverify foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_community\_relation;

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

/\* Table: t\_community\_relation \*/

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

create table t\_community\_relation

(

relation\_id bigint(5) not null,

user\_id bigint(5) not null,

doctor\_id bigint(5) not null,

primary key (relation\_id)

);

alter table t\_community\_relation add constraint FK\_fk\_doctor\_relation foreign key (doctor\_id)

references t\_community\_doctorverify (doctor\_id) on delete restrict on update restrict;

alter table t\_community\_relation add constraint FK\_fk\_user\_relation foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_community\_chat;

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

/\* Table: t\_community\_chat \*/

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

create table t\_community\_chat

(

chat\_id bigint(5) not null,

relation\_id bigint(5) not null,

chat\_time time not null,

chat\_content text not null,

chat\_from char(2) not null,

primary key (chat\_id)

);

alter table t\_community\_chat add constraint FK\_fk\_relation\_chat foreign key (relation\_id)

references t\_community\_relation (relation\_id) on delete restrict on update restrict;

drop table if exists t\_community\_post;

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

/\* Table: t\_community\_post \*/

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

create table t\_community\_post

(

post\_id bigint(10) not null,

user\_id bigint(5) not null,

user\_name varchar(20) not null,

post\_time time not null,

post\_title varchar(20) not null,

post\_content text not null,

post\_image blob,

like\_number bigint(5) not null,

favorite\_number bigint(5) not null,

primary key (post\_id)

);

alter table t\_community\_post add constraint FK\_fk\_user\_post foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_community\_interaction;

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

/\* Table: t\_community\_interaction \*/

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

create table t\_community\_interaction

(

interaction\_id bigint(5) not null,

post\_id bigint(10) not null,

user\_id bigint(5) not null,

is\_like boolean not null,

is\_favorite boolean not null,

primary key (interaction\_id)

);

alter table t\_community\_interaction add constraint FK\_fk\_post\_interaction foreign key (post\_id)

references t\_community\_post (post\_id) on delete restrict on update restrict;

alter table t\_community\_interaction add constraint FK\_fk\_user\_interaction foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_diet\_record;

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

/\* Table: t\_diet\_record \*/

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

create table t\_diet\_record

(

diet\_id bigint(5) not null,

food\_id bigint(5) not null,

user\_id bigint(5) not null,

diet\_date date not null,

diet\_type char(2) not null,

primary key (diet\_id)

);

alter table t\_diet\_record add constraint FK\_fk\_food\_diet foreign key (food\_id)

references t\_community\_food (food\_id) on delete restrict on update restrict;

alter table t\_diet\_record add constraint FK\_fk\_user\_diet foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_health\_medication;

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

/\* Table: t\_health\_medication \*/

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

create table t\_health\_medication

(

medication\_id bigint(5) not null,

user\_id bigint(5) not null,

drug\_id bigint(5) not null,

medication\_date date not null,

medication\_type char(2) not null,

is\_take boolean not null,

primary key (medication\_id)

);

alter table t\_health\_medication add constraint FK\_fk\_drug\_medication foreign key (drug\_id)

references t\_community\_drug (drug\_id) on delete restrict on update restrict;

alter table t\_health\_medication add constraint FK\_fk\_user\_medication foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_health\_record;

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

/\* Table: t\_health\_record \*/

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

create table t\_health\_record

(

health\_record\_id bigint(5) not null,

user\_id bigint(5) not null,

record\_date date not null,

user\_bloodsugar decimal(3,1) not null,

user\_low\_bloodpressure int not null,

user\_high\_bloodpressure int not null,

user\_heartrate int not null,

user\_weight int not null,

user\_health\_condition varchar(5) not null,

primary key (health\_record\_id)

);

alter table t\_health\_record add constraint FK\_fk\_user\_health foreign key (user\_id)

references t\_all\_user (user\_id) on delete restrict on update restrict;

drop table if exists t\_health\_suggest;

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

/\* Table: t\_health\_suggest \*/

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

create table t\_health\_suggest

(

suggest\_id bigint(5) not null,

health\_data\_type char(2) not null,

health\_data\_range varchar(10) not null,

suggest\_content text not null,

primary key (suggest\_id)

);

# 4 运用设计

## 4.1 数据字典设计

略。

## 4.2 安全设计

人体生理数据监测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。