**数据库原理与应用大作业**

## 题 目： A Home-Healthcare Information System

## 学 院： 计算机科学与技术学院

## 专 业： 计算机

## 班 级：

## 姓 名：

## 指导老师：

## 起讫日期： 2020.11.01-2020.12.13

一．对系统的理解

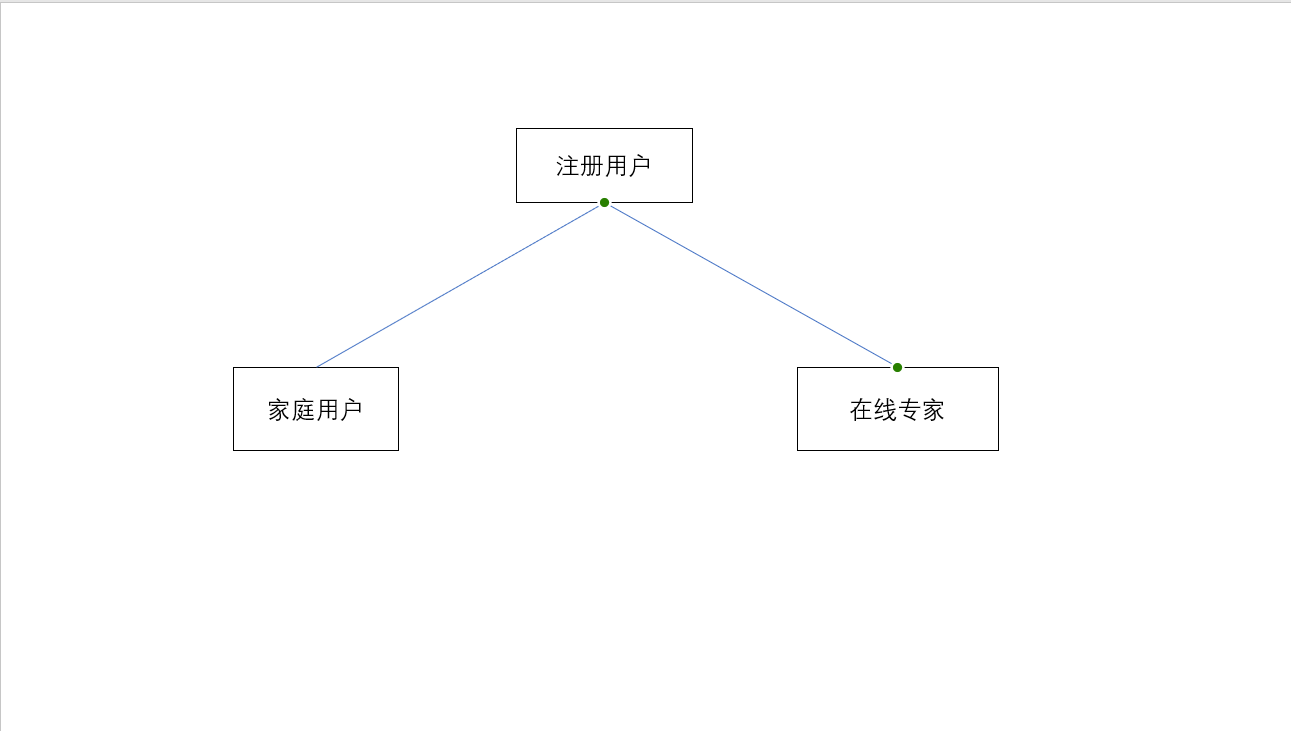
**本系统的主要功能；用户注册进入系统，根据用户类型不同登记用户的个人信息。**

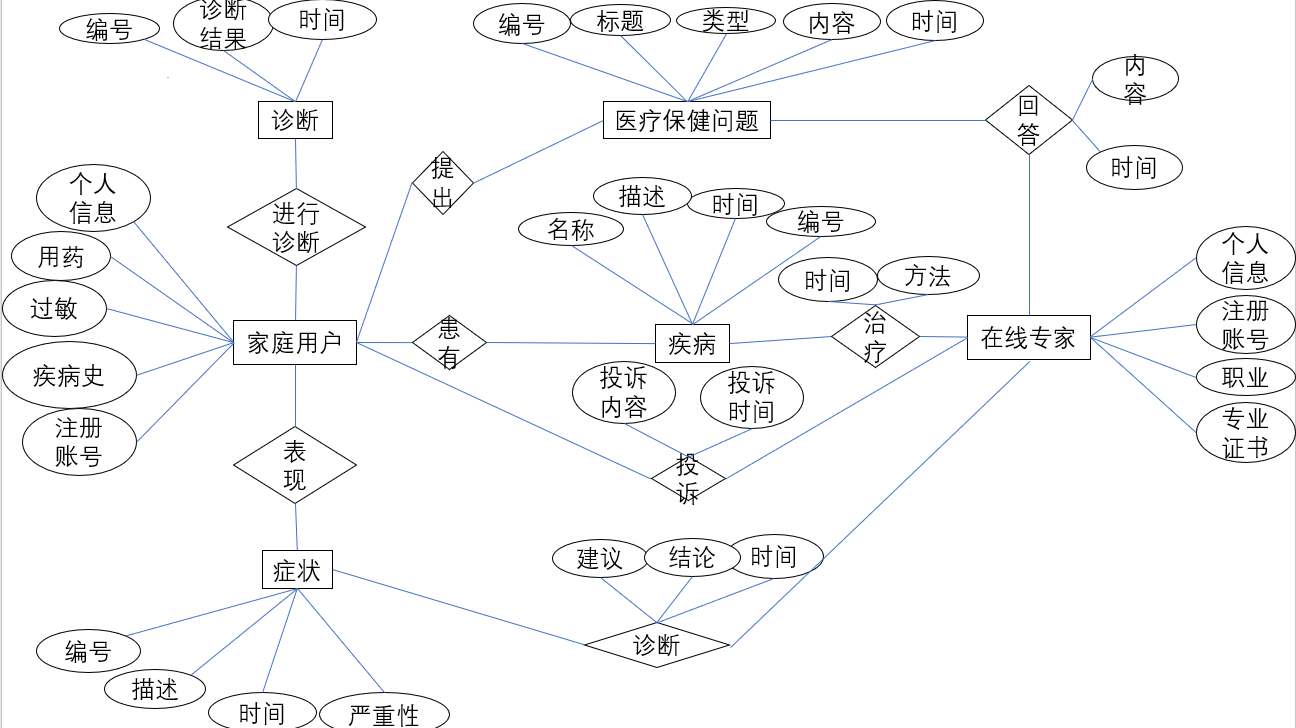
**家庭用户主要功能：输入健康状况，记录家庭诊断测试结果，提出健康问题。可以查看在线专家的诊断，治疗建议和健康问题的回答。**

**在线专家根据家庭用户的健康状况和测试结果给出相应的诊断，治疗建议，在线专家可以回答多个家庭用户提出的健康问题，但是如果问题类型为私人的话只能由私人在线专家回答。**

**系统管理员可以手动添加，修改，删除用户的信息。查询用户表和更新表信息等。**

二．E\_R图





三．表结构

注册信息表 user\_ informationOfregistration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为主键（否无需填写） | 是否可以为空 | 文字说明 |
| id | Numeric(10) | 是 | Not null | 注册账号 |
| phone | Numeric(11) |  | Not null | 联系方式 |
| name | Varchar(20) |  | Not null | 姓名 |
| place | Varchar(50) |  | Not null | 地址 |
| sex | Varchar(2) |  |  | 性别 |
| age | Numeric(3) |  |  | 年龄 |

家庭用户表 home\_users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为主键（否无需填写） | 是否可以为空 | 文字说明 |
| User\_id | Numeric(10) | 是 | Not null | 注册的账号 |
| Current\_drugs | Varchar(50) |  |  | 用户当前用药 |
| Disease\_history | Varchar(50) |  |  | 用户的疾病史 |
| Allergy | Varchar(50) |  |  | 用户过敏的说明 |
| prvivatedoctor\_id | Numeric(10) |  |  | 私人医生的id |

在线专家表 online\_experts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为主键（否无需填写） | 是否可以为空 | 文字说明 |
| Expert\_id | Numeric(10) | 是 | Not null | 专家的账号 |
| Professin | Varchar(20) |  | Not null | 专家的职业类型 |
| Credentials | Varchar(100) |  | Not null | 专家的证书 |

症状表 symptoms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为主键（否无需填写） | 是否可以为空 | 文字说明 |
| Sy\_id | Number(10) | 是 | Not null | 症状编号 |
| User\_id | Number(10) |  | Not null | 拥有此症状人id |
| Sy\_ describe | Varchar(100) |  | Not null | 症状描述 |
| Sy\_severity | Varchar(10) |  | Not null | 症状严重性 |
| Sy\_time | date |  | Not null | 症状时间 |

家庭用户诊断测试表 Diagnostic\_test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为主键（否无需填写） | 是否可以为空 | 文字说明 |
| Test\_id | Number(10) | 是 | Not null | 诊断测试的id |
| User\_id | Number(10) | 外键 | Not null | 被诊断人的id |
| Test\_describe | Varchar(100) |  | Not null | 诊断结果 |
| Test\_time | Date |  | Not null | 诊断时间 |

病例表disease

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 是否为主键（否无需填写） | 是否可以为空 | 文字说明 |
| Disease\_id | Numeric(10) | 是 | Not null | 疾病编号 |
| Disease\_name | Varchar(20) |  | Not null | 疾病名称 |
| Disease\_description | Varchar(50) |  | Not null | 疾病描述 |
| Disease\_time | Date |  | Not null | 疾病检测时间 |
| User\_id | Numeric(10) | 外键 |  | 患者id |

疾病治疗表treat \_disease

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 主键 | 是否允许为空 | 数据说明 |
| Disease\_id | Number(10) | 是 | Not null | 疾病编号id |
| Expert\_id | Number(10) | 外键 | Not null | 专家id |
| Treat\_way | Varchar2(100) |  | Not null | 治疗方法 |
| Treat\_time | Date |  | Not null | 治疗时间 |

医疗保健问题 Health\_questions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 主键 | 是否允许为空 | 数据说明 |
| Hq\_id | Number(10) | 是 | Not null | 问题编号 |
| Hq\_name | Varchar(20) |  | Not null | 问题标题 |
| User\_id | Number(10) |  | Not null | 问问题的人 |
| Hq\_type | Varchar(2) |  | Not null | 问题类型私人或者开放 |
| Hq\_content | Varchar(100) |  | Not null | 问题内容 |
| Hq\_start | Date |  | Not null | 提问时间 |

回答医疗保健问题 answers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 主键 | 是否允许为空 | 数据说明 |
| Hq\_id | Number(10) | 是 | Not null | 问题编号 |
| Expert\_id | Number(10) | 是 | Not null | 回答的专家 |
| Answer | Varchar(100) |  | Not null | 回答的内容 |
| Answer\_time | Date |  | Not null | 回答时间 |

投诉 complaints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 主键 | 是否允许为空 | 数据说明 |
| User\_id | Number(8) | 是 | Not null | 投诉用户 |
| Expert\_id | Number(8) | 是 | Not null | 投诉专家 |
| Com\_time | date |  | Not null | 投诉时间 |
| Com\_content | Varchar(100) |  | Not null | 投诉描述 |

**四．Sql建表**

Create table User\_informationOfregistration

(

id Numeric(10) not null,

phone Numeric(11) Not null,

name Varchar(20) Not null,

place Varchar(50) Not null,

sex Varchar(2),

age Numeric(3),

primary key(id)

);

Create table home\_users

(

User\_id Numeric(10) Not null,

Current\_drugs Varchar(50),

Disease\_history Varchar(50),

Allergy Varchar(50),

prvivatedoctor\_id Numeric(10),

primary key(User\_id)

);

Create table online\_experts

(

Expert\_id Numeric(10) Not null ,

Expert\_name Varchar(20) Not null ,

Expert\_phone Numeric(11) Not null ,

Professin Varchar(20) Not null ,

Credentials Varchar(100) Not null ,

Primary key(Expert\_id)

);

Create table symptoms

(

Sy\_id Number(10) not null,

User\_id Number(10) Not null,

Sy\_ describe Varchar(100) Not null,

Sy\_severity Varchar(10) Not null,

Sy\_time date Not null,

Primary key(Sy\_id)

);

Create table Diagnostic\_test

(

Test\_id Number(10) Not null ,

User\_id Number(10) Not null ,

Test\_describe Varchar(100) Not null ,

Test\_time Date Not null ,

Primary key(Test\_id)

);

Create table disease

(

Disease\_id Numeric(10) Not null ,

Disease\_name Varchar(20) Not null ,

Disease\_description Varchar(50) Not null ,

Disease\_time Date Not null ,

User\_id Numeric(10) Not null ,

Primary key(Disease\_id)

);

Create table treat \_disease

(

Disease\_id Number(10) Not null ,

Expert\_id Number(10) Not null ,

Treat\_way Varchar2(100) Not null ,

Treat\_time Date Not null ,

Primary key(Disease\_id)

);

Create table Health\_questions

(

Hq\_id Number(10) Not null,

Hq\_name Varchar(20) Not null,

User\_id Number(10) Not null,

Hq\_type Varchar(2) Not null,

Hq\_content Varchar(100) Not null ,

Hq\_strat Date Not null ,

Primary key(Hq-id)

);

Create table answers

(

Hq\_id Number(10) Not null ,

Expert\_id Number(10) Not null ,

Answer Varchar(100) Not null ,

Answer\_time Date Not null ,

Primary key(Hq\_id,Expert\_id)

);

Create table complaints

(

User\_id Number(8) Not null ,

Expert\_id Number(8) Not null ,

Com\_time date Not null ,

Com\_content Varchar(100) Not null ,

Primary key(User\_id,Expert\_id)

);

五.功能查询

1. 简单查询（3个）
2. 查询一个名字带有“BA”的所有个人身份信息；

Select \*

From user\_ informationOfregistration

Where name like’%B%’;

1. 查询一个叫“A”的人的疾病史；

Select Disease\_history

From home\_users

Where User\_id=(select id from user\_ informationOfregistration where name=’A’);

1. 查询一个叫id为00001的在线专家的所有回答的内容；

Select answer

From answers

Where Expert\_id=’000001’;

1. 连接查询（5个）
2. 查询名字叫“A”的人的所有医疗保健问题的标题和内容。

Select Hq\_id , Hq\_content

From Health\_questions join home\_users

On home\_users.User\_id= Health\_questions.user\_id

AND home\_users.User\_name=’A’;

1. 查询年龄为20的用户的患有疾病的人的姓名和疾病名称；

Select user\_ informationOfregistration.name,disease.name

From user\_informationOfregistration join disease

On disease.User.id= user\_ informationOfregistration.id

Where user\_ informationOfregistration.age=20;

1. 查询所有投诉中投诉人姓名,被投诉的次数。

Select user\_ informationOfregistration.name , count(complaint.com\_content)

From user\_ informationOfregistration join complaints

on complaints.Expert\_id= user\_ informationOfregistration.id

Group by complaints.Expert\_id;

1. 查询所有年龄20的家庭用户的姓名，症状描述，时间，（没有症状的也把姓名写出来；

Select user\_ informationOfregistration.name , Sy\_ describe , Sy\_time

From user\_ informationOfregistration left join symptoms

On =symptoms.sy\_id =user\_ informationOfregistration

Where user\_ informationOfregistration.age=20;

1. 查询患有疾病的人的姓名,所患疾病，治疗方法；

Select user\_ informationOfregistration.name ,treat\_disease. treat \_way

From (disease join treat \_disease On =disease.Disease\_id=treat\_disease.Disease\_id)

Join user\_ informationOfregistration

On user\_ informationOfregistration.id= disease.user\_id;

1. 存储过程（5个）
2. 所有使用者年龄加一；

Create or replace procedure add\_age

As

Begin

Update user\_ informationOfregistration

Set user\_ informationOfregistration.age= user\_ informationOfregistration.age+1;

End;

1. 删除编号为100001的在线专家的投诉

Create or replace procedure delete\_complaint(num in complaints.Expert\_id%type)

as

Begin

Delete

From complaints

Where complaints.Expert\_id=num;

End;

1. 记录一下user\_id1的诊断结果

Create or replace procedure add\_ Diagnostic(test\_id1 in Diagnostic .Test\_id%type,

User\_id1 in Diagnostic.User\_id%type ,content in Diagnostic .Test\_describe%type)

As

Time1 date

Time1=now()

Begin

Insert into Diagnostic\_test(Test\_id,User\_id,Test\_describe,Test\_time

Values(test\_id1,User\_id1,content,Time1);

End;

1. 查询名字中带有“A”的姓名的用户（包括在线专家）存储过程

Create or replace procedure find\_user

As

Name2 user\_ informationOfregistration.name%type;

Begin

Select name into Name2

From user\_ informationOfregistration

Where name like ‘%a%’;

End;

1. 向answer表中A回答的编号为B的问题 回答内容为C；

Create or replace procedure update\_answer

(A in answers.Expert\_id%type,

B in answers.Hq\_id%type,

C in answers.answer%type,

D in answers.answer\_time)

As

Begin

Insert into answers

Values(A,B,C,D)

End;

4.触发器（3个）

1.删除有症状的记录的用户信息，会重新添加删除的数据

Create or replace trigger t1

Before delete on user\_ informationOfregistration

For each row

Num number(5)

Begin

Selete count(\*) into num

From user\_ informationOfregistration, symptoms

Where user\_ informationOfregistration.id= symptoms.user\_id;

If total>0 then

Insert into user\_ informationOfregistration(id,phone,name,place,sex,age)

Values(:old.id, :old.phone, :old.name, :old.place, :old.sex, :old.age);

End if;

End;

2. 当多出一个用户的时候给出提示

Create or replace trigger t2

After insert on user\_ informationOfregistration

For each row

Begin

DBMS\_output.put\_line(‘注册成功‘)；

End;

3,当修改个人信息是给出提示

Create or replace trigger t3

After update on user\_ informationOfregistration

For each row

Begin

DBMS\_output.put\_line(‘修改成功 ‘)；

End;

**六．总结**

我们的数据库大作业终于完成了，但是，自己离真正熟练掌握数据库还有很长的距离。

我们这次大作业是家庭医疗管理系统。就从看题目开始就已经很困难了，用百度翻译的话，不能很好的准确理解题目的意思，所以我就一边翻译，一边尝试 着看那一大篇的英语文章。

回顾这个做作业的过程，我们一直在查阅资料，所以说我们并没有很好的学好并记住sql语言，但是我相信，我们日积月累，还有什么做不好的呢？

如果只是一味的学习而不懂得实践，那也是不行的。通过这次独立的建立一个家庭医疗信息系统的关系数据库，从将现实世界的对象转换成实体找出联系，画出E-R图，再将E-R图转换成关系模式，再对关系模式进行优化，转化成3NF，这样极大的减少基本表的数据冗余和对数据的操作困难。然后进行建表，我才用所有的数据操作都使用存储过程来实现，这样能极大的提高效率。创建视图，游标，函数，触发器以及存储过程来实现各种功能，对数据库的基本操作和理解加深了很多。

