数据库原理与应用大作业家庭保健信息系统计算机科学与技术学院

姓			名:	
学			号:	
专			业:	
班			级:	
指	导	教	师:	

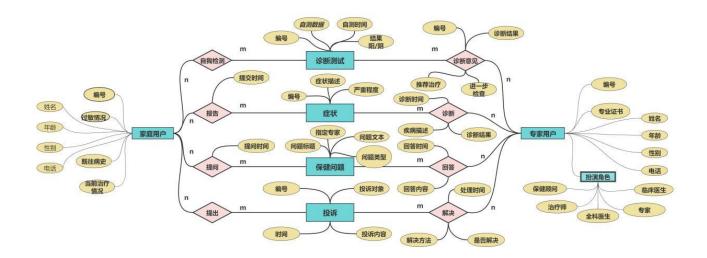
一、系统描述

1. 需求分析

系统中存在两种用户——家庭用户和在线专家。家庭用户信息包括联系方式、个人信息、病史、当前正在使用的药物或进行的治疗、过敏信息。每个家庭用户可以有一个专门的在线专家负责健康咨询,报告症状,提交诊断自测结果以及对专家进行投诉等。在线专家信息包括联系方式,个人信息,专业证书,扮演角色等。在线专家可以通过该系统对医疗保健问题进行回答,对症状报告进行诊断,对诊断自测报告给出建议以及处理并解决来自用户的投诉。两种用户之间存在这样的关系:

- 1. 附件 A 参考表(test): 不用用户填写,用于给家庭用户进行家庭诊断自测的时候参考,包含以下的信息: 参考编号,自测条件,自测过程,自测工具,自测方式。
- 2. 自我检测表(test_user):由家庭用户经过诊断自测后填写,包含以下的信息:自我检测编号,参考编号,用户编号,自测数据,自测时间,自测结果。
- 3. 专家诊断意见表(test_expert): 由专家阅读诊断自测后填写,包含以下的信息: 专家诊断意见编号,自我检测编号,专家编号,诊断结果,推荐治疗,进一步检查意见。
- 4. 症状报告表(report_user):由家庭用户填写,用于报告用户的身体症状,可包含以下的信息:症状报告编号,报告者(家庭用户编号),报告时间,症状描述,严重程度。
- 5. 症状诊断表(report_expert):由在线专家根据用户的症状报告填写,可包含以下的信息:症状诊断编号,专家编号,症状报告编号,诊断时间,诊断结果,病情描述。
- 6. 保健问题表(health_user):有家庭用户填写,可包以下信息,问题表编号,用户编号,提问时间,提问标题,问题类型(公开/私有),问题内容,指定提问专家。
- 7. 保健问题回答表(health_expert):由在线专家填写,可包含以下信息:回答表编号,保健问题编号,专家编号,回答时间 ,回答内容。
- 8. 投诉表(complaint_user): 由家庭用户填写,对专家进行投诉,可包含以下的信息: 投诉表编号,家庭用户编号,投诉对象(专家编号),投诉时间,投诉内容。
- 9. 投诉反馈表(complaint_expert):由被投诉的专家填写,包含以下信息:反馈表编号,处理人(专家编号),投诉表编号,解决时间,处理办法,是否解决。

2. 整体 E-R 图



二、系统基础表设计

1. 表设计

1) 家庭用户表(USER)

字段名	数据类型	含义	备注
user_id	INT	用户 ID	主键,自增
user_password	VARCHAR(128)	登录密码哈希	不能为空
username	VARCHAR(50)	用户名称	不能为空
user_age	INT	用户年龄	
user_sex	TINYINT	用户性别	0为女,1为男
user_phone	VARCHAR(11)	用户电话	
is_allergic	TINYINT(1)	是否过敏	0 为无, 1 为有
medical_history	TEXT	既往病史	
current_treatment	TEXT	当前治疗情况	

2) 医生专家信息表(EXPERT)

字段名	数据类型	含义	备注
expert_id	INT	在线专家 ID	主键,自增
expert_password	VARCHAR(128)	登录密码哈希	
expert_name	VARCHAR(50)	专家名字	不能为空
expert_age	INT	专家年龄	0 为女, 1 为男
user_sex	TINYINT	专家性别	0 为女, 1 为男
expert_phone	VARCHAR(11)	专家电话	
professional_honor	VARCHAR(50)	专业证书	不能为空

3) 扮演角色表(ROLE)

字段名	数据类型	含义	备注
role_id	INT	扮演角色 ID	主键,自增
role_name	VARCHAR(50)	扮演角色名字	

4) 专家与角色关系表(EXPERT_ROLE)

字段名	数据类型	含义	备注
role_id	INT	扮演角色 ID	主键
expert_id	INT	专家 ID	主键

5) 附件 A 参考表(TEST)

字段名	数据类型	含义	备注
test_id	INT	附件 A 参考 ID	主键,自增
test_condition	TEXT	自测条件	
test_process	TEXT	自测过程	
test_tool	TEXT	自测工具	
test_method	TEXT	自测方式	

6) 自我检测表(TEST_USER)

字段名	数据类型	含义	备注
utest_id	INT	自我检测 ID	主键,自增
test_id	INT	附件 A 参考 ID	外键
user_id	INT	用户 ID	外键
result_data	TEXT	自测数据	
result_time	TIMESTAMP	检测时间	默认当前时间

result	CHAR(1)	结果状态	肾性. 肾性
Tesuit		汨木巛心	71 II.

7) 专家诊断意见表(TEST_EXPERT)

字段名	数据类型	含义	备注
etest_id	INT	专家诊断意见 ID	主键,自增
utest_id	INT	自我检测 ID	外键
expert_id	INT	专家 ID	外键
etest_time	TIMESTAMP	诊断时间	默认当前时间
etest_data	TEXT	诊断结果	
etest_treat	TEXT	推荐治疗	
etest_survey	TEXT	进一步检查	

8) 症状报告表(REPORT_USER)

字段名	数据类型	含义	备注
ureport_id	INT	症状报告 ID	主键,自增
user_id	INT	用户 ID	外键
report_time	TIMESTAMP	检测时间	默认当前时间
detail	TEXT	症状描述	
degree	TEXT	严重程度	

9) 症状诊断表(REPORT_EXPERT)

字段名	数据类型	含义	备注
ereport_id	INT	症状诊断告 ID	主键,自增
expert_id	INT	专家 ID	外键
ureport_id	INT	症状报告 ID	外键
treat_time	TIMESTAMP	诊断时间	默认当前时间
result	VARCHAR(100)	诊断结果	
discribe	TEXT	病情描述	

10) 保健问题表(HEALTH_USER)

字段名	数据类型	含义	备注
question_id	INT	保健问题 ID	主键,自增
user_id	INT	用户 ID	外键
question_time	TIMESTAMP	提问时间	默认当前时间
question_name	VARCHAR(20)	问题名字	
question_type	char(2)	问题类型	公开,私有

question_content	TEXT	问题文本	
choice_expert	VARCHAR(50)	指定提问专家	默认为空

11)保健问题回答表(HEALTH_EXPERT)

字段名	数据类型	含义	备注
answer_id	INT	回答 ID	主键,自增
question_id	INT	保健问题 ID	外键
expert_id	INT	专家 ID	外键
answer_time	TIMESTAMP	回答时间	默认当前时间
answer_content	TEXT	回答内容	

12) 投诉表(COMPLAINT_USER)

字段名	数据类型	含义	备注
com_id	INT	投诉 ID	主键,自增
user_id	INT	用户 ID	外键
expert_id	INT	投诉对象(专家 ID)	外键
com_time	TIMESTAMP	投诉时间	默认当前时间
com_content	TEXT	投诉内容	

13) 投诉反馈表(COMPLAINT_EXPERT)

字段名	数据类型	含义	备注
solve_id	INT	反馈 ID	主键,自增
expert_id	INT	处理人(专家 ID)	外键
com_id	INT	投诉 ID	外键
solve_time	TIMESTAMP	反馈时间	默认当前时间
solve	TEXT	处理办法	
is_solve	boolean	是否解决	默认为 FALSE

2. SQL 建表 (包括索引、完整性约束)

创建数据库

使用 utf8mb4 字符集,使用utf8mb4_unicode_ci 排序规则

CREATE DATABASE home_healthcare

CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

创建家庭表

CREATE TABLE USER(

user_id INT auto_increment PRIMARY KEY,

```
user_password VARCHAR(128) NOT NULL,
 username VARCHAR(50) not NULL,
 user_age INT,
 user_sex TINYINT CHECK (user_sex IN (0,1)),
 user_phone VARCHAR(11),
 is_allergic TINYINT(1) DEFAULT 0 COMMENT '是否过敏, 0为无, 1为有',
 medical_history TEXT COMMENT '既往病史',
 current_treatment TEXT COMMENT '当前治疗情况'
);
# 创建医生信息表
CREATE TABLE expert(
 expert_id INT auto_increment PRIMARY KEY,
 expert password VARCHAR(128) NOT NULL,
 expert_name VARCHAR(50) NOT NULL,
 expert age INT,
 expert_sex TINYINT(1) CHECK(expert_sex IN (0,1)),
 expert phone VARCHAR(11),
 Professional_honor VARCHAR(50) NOT NULL COMMENT '专业证书'
);
# 角色表
CREATE TABLE role(
 role_id INT auto_increment PRIMARY KEY,
 role name VARCHAR(50) NOT NULL COMMENT '扮演角色'
);
# 插入五个角色
INSERT INTO role(role_name) VALUES ('保健顾问'),('治疗师'),('全科医生'),('专家
'),('临床医生');
# 专家角色关系表
CREATE TABLE expert_role(
 role id INT NOT NULL,
 expert id INT NOT NULL,
 PRIMARY KEY (expert_id, role_id),
 FOREIGN KEY (expert_id) REFERENCES expert(expert_id),
 FOREIGN KEY (role_id) REFERENCES role(role_id)
);
# 附件A参考表
CREATE TABLE test(
 test_id INT auto_increment PRIMARY KEY,
 test_condition TEXT COMMENT '自测条件',
```

```
test_process TEXT COMMENT '自测过程',
 test_tool TEXT COMMENT '自测工具',
 test_method TEXT COMMENT '自测方式'
);
# 自我检测表
CREATE table test_user(
 utest_id INT auto_increment PRIMARY KEY, # 主键
 test id INT NOT NULL, # 测试参考表
 user id INT NOT NULL, # 用户表
 result data TEXT COMMENT '自测数据',
 result_time TIMESTAMP,
 result char(1) CHECK (result IN('阴','阳')),
 FOREIGN KEY (test id) REFERENCES test(test id),
 FOREIGN KEY (user_id) REFERENCES `user`(user_id)
);
# 专家诊断意见表
CREATE table test_expert(
 etest_id INT auto_increment PRIMARY KEY,
 utest id INT NOT NULL, #回答的自我诊断
 expert id INT NOT NULL,# 专家
 etest time TIMESTAMP,
 etest_data TEXT COMMENT '诊断结果',
 etest treat TEXT COMMENT '推荐治疗',
 etest survey TEXT COMMENT '进一步检查',
 FOREIGN KEY (utest_id) REFERENCES test_user(utest_id),
 FOREIGN KEY (expert_id) REFERENCES expert(expert_id)
);
# 症状报告表
CREATE table report_user(
 ureport_id INT auto_increment PRIMARY KEY,
 user_id INT NOT NULL, # 报告者
 report time TIMESTAMP, # 报告时间
 detail TEXT COMMENT'症状描述',
 degree VARCHAR(20) COMMENT '严重程度',
 FOREIGN KEY (user_id) REFERENCES `user`(user_id)
);
# 症状诊断表
CREATE table report_expert(
 ereport_id INT auto_increment PRIMARY KEY,
 expert id INT NOT NULL,
```

```
ureport_id INT NOT NULL,
 treat time TIMESTAMP,
 result VARCHAR(100) COMMENT'诊断结果',
 discribe TEXT COMMENT'病情描述',
 FOREIGN KEY (expert_id) REFERENCES expert(expert_id),
 FOREIGN KEY (ureport_id) REFERENCES report_user(ureport_id)
);
# 保健问题表
CREATE table health_user(
 question id INT auto increment PRIMARY KEY,
 user id INT NOT NULL,
 question_time TIMESTAMP,
 question name VARCHAR(20) COMMENT '问题名字',
 question_type char(2) CHECK(question_type IN ('公开','私有')),
 question content TEXT COMMENT '问题文本',
 choice expert VARCHAR(50) DEFAULT NULL, # 指定提问专家
 FOREIGN KEY (user id) REFERENCES `user`(user id)
);
# 保健问题回答表
CREATE table health_expert(
 answer id INT auto increment PRIMARY KEY,
 question_id INT NOT NULL, # 保健问题
 expert id INT NOT NULL, # 回答的专家
 answer_time TIMESTAMP,
 answer_content TEXT,
 FOREIGN KEY (question id) REFERENCES health user(question id),
 FOREIGN KEY (expert_id) REFERENCES expert(expert_id)
);
# 投诉表
CREATE table complaint user(
 com_id INT auto_increment PRIMARY KEY,
 user id INT NOT NULL,
 expert_id INT COMMENT'投诉对象',
 com time TIMESTAMP,
 com_content TEXT COMMENT '投诉内容',
 FOREIGN KEY (user_id) REFERENCES `user`(user_id),
 FOREIGN KEY (expert id) REFERENCES expert(expert id)
);
# 投诉反馈表
CREATE table complaint_expert(
```

```
solve_id INT auto_increment PRIMARY KEY,
expert_id INT NOT null COMMENT '处理人',
com_id INT COMMENT '投诉表',
solve_time TIMESTAMP,
solve TEXT COMMENT '处理办法',
is_solve boolean DEFAULT FALSE COMMENT '是否解决'
);
```

三、系统功能 SQL 实现

(在本节中,需要同学们使用 SQL 和 PL/SQL 完成多种自定义功能的实现,并将结果截图保存。具体要求: 3 个简单查询; 3 个多表查询; 2 个视图; 1 个游标; 1 个自定义函数; 2 个存储过程; 2 个触发器)

1. 简单查询

1) 家庭用户信息(user_info)

-- 家庭用户<u>信息查询</u>,得到家庭用户的一些基本信息、是否过敏、既往病史、当前治疗情况

SELECT * FROM `user`;

user_id	user_password	username	user_age	user_sex	user_phone	is_allergic	medical_history	current_treatment
1	123	张三	19	1	12345678901	0	无	就诊吃药
2	123	李四	22	0	12345678901	1	花粉过敏	定期复查
3	123	王五	28	1	12345678901	1	无	无
4	123	孙七	31	0	44455566677	1	青霉素过敏	正在接受抗生素治疗
5	123	王十一	21	0	1122334455	0	无	正在接受物理治疗

2) 专家信息 (expert_info)

-- 专家年龄在 40 岁及以上的医生 将 1 映射为男性,将 0 映射为女性并按年龄排序

SELECT

```
e.expert_id AS 'ID', e.expert_name AS '姓名', e.expert_age AS '年龄', CASE
```

```
WHEN e.expert_sex = 1 THEN '男'
WHEN e.expert_sex = 0 THEN '女'
ELSE '未知'
```

END AS 性别,

e.expert_phone AS '电话', e.Professional_honor AS '专业证书'
FROM expert e WHERE e.expert_age >= 40 ORDER BY e.expert_age DESC;

expert_id	expert_password	expert_name	expert_age	expert_sex	expert_phone	Professional_honor
1	123	陈医生	45	1	13900001112	内科专家
2	123	李医生	38	0	13800002233	外科专家
3	123	王医生	52	1	13700003344	儿科专家
4	123	赵医生	40	0	13600004455	眼科专家

ID	姓名	年龄	性别	电话	专业证书
3	王医生	52	男	13700003344	儿科专家
5	孙医生	48	男	13500005566	心脏科专家
1	陈医生	45	男	13900001112	内科专家
4	赵医生	40	女	13600004455	眼科专家

3) 过敏情况统计 (allergy_statistics)

-- 将家庭用户人员 **30 岁以下**的人按**是否过敏**分类

并<u>统计</u>元组个数、年龄<u>最大值</u>、<u>最小值</u>、<u>平均值</u>、年龄<u>求和</u>

SELECT COUNT(user_id) 个数, MAX(user_age) 最大值, MIN(user_age) 最小值, AVG(user_age) 平均值, SUM(user_age) 和, CASE

WHEN is_allergic = 0 THEN

'不是' ELSE '是'

END AS 是否过敏

FROM `user` WHERE user_age < 30 GROUP BY is_allergic HAVING is_allergic = 0;</pre>

user_id	user_password	username	user_age	user_sex	user_phone	is_allergic	$medical_history$	current_treatment
1	123	张三	19	1	12345678901	0	无	就诊吃药
2	123	李四	22	0	12345678901	1	花粉过敏	定期复查
3	123	王五	28	1	12345678901	1	无	无
4	123	孙七	31	0	44455566677	1	青霉素过敏	正在接受抗生素治疗
5	123	王十一	21	0	1122334455	0	无	正在接受物理治疗

	信息								
个数		最大值	最小值	平均值	和	是否过敏			
	2	21	19	20.0000	40	0			
	2	28	22	25.0000	50	1			

 数据
 信息

 个数
 最大值
 最小值
 平均值
 和
 是否过敏

 2
 21
 19
 20.0000
 40
 不是

2. 多表查询

1) 专家完整信息(expert_all_info)

-- 专家完整信息,处理三个表展现专家的个人信息、专业证书、扮演角色 SELECT expert_id ID, expert_name 专家名字, expert_age 年龄, CASE

WHEN expert_sex = 1 THEN '男' ELSE '女'

END 性别, expert_phone 电话, Professional_honor 专业证书,role_name 扮演角色 FROM expert NATURAL JOIN expert_role NATURAL JOIN role;

role_id	expert_id	expert_password	expert_name	expert_age	expert_sex	expert_phone	Professional_honor	role_nam
1	1	123	陈医生	45	1	13900001112	内科专家	保健顾问
2	2	123	李医生	38	0	13800002233	外科专家	治疗师
3	3	123	王医生	52	1	13700003344	儿科专家	全科医生
4	4	123	赵医生	40	0	13600004455	眼科专家	专家
5	5	123	孙医生	48	1	13500005566	心脏科专家	临床医生

ID	专家名字	年龄	性别	电话	专业证书	扮演角色
1	陈医生	45	男	13900001112	内科专家	保健顾问
2	李医生	38	女	13800002233	外科专家	治疗师
3	王医生	52	男	13700003344	儿科专家	全科医生
4	赵医生	40	女	13600004455	眼科专家	专家
5	孙医生	48	男	13500005566	心脏科专家	临床医生

2) 查询投诉有关信息(complaint_detail)

-- 查询投诉信息,包含四个表的处理,展现投诉关系中的投诉者、投诉内容、处理人、解决办法、解决结果

SELECT c.com_id ID, u.username 投诉者, c.com_time 投诉时间,c.com_content 投诉内容,e.expert_name 投诉对象,e.expert_name 处理人,ce.solve_time 解决时间,ce.solve 解决办法,ce.is_solve 是否解决

from complaint_user c

JOIN `user` u ON c.user_id = u.user_id

JOIN expert e ON c.expert_id = e.expert_id

JOIN complaint_expert ce ON ce.com_id = c.com_id;

ID	投诉者	投诉时间	投诉内容	投诉对象	处理人	解决时间	解决办法	是否解决
1	张三	2024-05-01 15:20:00	我对专家的服务?	李医生	李医生	2024-11-29 11:19:43	已与用户沟通,承诺改进	1
2	王五	2024-11-29 11:15:33	专家给出的诊断结	赵医生	赵医生	2024-11-29 11:19:43	提供了更详细的诊断报告	1
3	王 +-	2024-11-29 11:15:33	我认为专家的回答	陈医生	陈医生	2024-11-29 11:19:43	安排了另一位专家进行二次	0
4	李四	2024-11-29 11:15:33	专家没有按时回复	孙医生	孙医生	2024-11-29 11:19:43	已调整回复流程,确保及区	1
5	孙七	2024-11-29 11:15:33	我对专家推荐的消	王医生	王医生	2024-11-29 11:19:43	已提供第二意见,并与用户	1

3) 用户自我检测 (self_test)

-- 用户自测信息查询,三表连接嵌套查询,呈现用户检测的工具、方法、条件、结果

SELECT u.user_id ID,u.username 名字,tu.result_time 自测时间,t.test_tool 自测工具,t.test_method 自测方法, t.test_condition 自测条件,tu.result 自测结果from `user`u JOIN test_user tu ON tu.user_id = u.user_id JOIN test t ON t.test_id = tu.test_id

WHERE EXISTS (SELECT * FROM test_user WHERE test_user.user_id = u.user_id);

ID	名字	自测时间	自测工具	自测方法	自测条件	自测结果
1	张三	2024-11-29 10:43:17	MyAllergyTest套件	皮肤针刺,观察反应	过敏症状	阳
1	张三	2024-11-29 10:43:17	AWARE™乳腺自检垫和套	自我检查,减少摩擦,提	乳腺癌自我检查	阴
2	李四	2024-11-29 10:43:17	Uri-Strip K试纸	试纸浸入尿液, 观察颜色:	血糖、蛋白、pH值(尿三	阴
3	王五	2024-11-29 10:43:17	EZ DETECT™测试垫	测试垫放置马桶,观察颜	大肠癌隐血筛查	阴
4	孙七	2024-11-29 10:41:11	LANDMARK	皮肤针刺,观察反应	前列腺癌PSA筛查	阴

3. 建立视图

1) 专家详情 (expert_detail)

-- 专家详情视图,用于向用户展示专家姓名、电话、专业资质、角色

CREATE VIEW view expert detail

AS

SELECT expert_id ID,expert_name, expert_age,expert_phone,Professional_honor,role_name

FROM expert NATURAL JOIN expert role NATURAL JOIN role;

ID	expert_name	expert_age	expert_phone	Professional_honor	role_name
1	陈医生	45	13900001112	内科专家	保健顾问
2	李医生	38	13800002233	外科专家	治疗师
3	王医生	52	13700003344	儿科专家	全科医生
4	赵医生	40	13600004455	眼科专家	专家
5	孙医生	48	13500005566	心脏科专家	临床医生

2) 症状详情 (symptom_detail)

-- 诊断详情视图,用于整体呈现用户的名字、症状描述、严重程度、诊断医生、医生的诊断结果,并且描述

CREATE VIEW view_diagnosis_detail

AS

SELECT ureport_id ,username ,report_time, detail, expert_name,result, degree,treat_time, discribe FROM report_user NATURAL JOIN `user` NATURAL JOIN report_expert NATURAL JOIN expert;



4. 游标

1) 男女专家分类 (expert_gender)

-- 男女专家分类,根据专家的性别,将他们分别插入到不同的新表中,包括名字,性别,专家 特长信息 delimiter // **CREATE PROCEDURE** cur_sex() **DECLARE** cur name **VARCHAR**(50); **DECLARE** sex **CHAR**(1); **DECLARE** pro honour **VARCHAR**(50); **DECLARE** done **INT DEFAULT** 0; DECLARE curl CURSOR FOR SELECT expert name, expert sex, Professional honor FROM expert; DECLARE CONTINUE HANDLER FOR NOT found set done = 1; **CREATE TABLE** cur male(m id INT auto increment PRIMARY KEY, m name VARCHAR(50), m sex CHAR(1), m pro VARCHAR(50)); **CREATE TABLE** cur female(fe id INT auto increment PRIMARY KEY, fe name VARCHAR(50), fe sex char(1), fe pro VARCHAR(50)); OPEN curl; read loop: LOOP FETCH curl INTO cur name, sex, pro honour; IF done = 1 THEN **LEAVE** read loop; END IF; IF sex = 1 THEN INSERT INTO cur male (m name, m sex, m pro) VALUES (cur name, '男', pro honour);

```
ELSE
        INSERT INTO cur female(fe name,fe sex,fe pro) VALUES (cur name,'女',pro honour);
     END IF;
  END LOOP;
  CLOSE curl;
END;
//
delimiter;
-- 案列: 对现有的 3 名男专家, 2 名女专家自行分类
CALL cur sex();
 m id
         m name
                     m sex
                               m_pro
                                            fe_id
                                                      fe_name
                                                                     fe_sex
                                                                                fe_pro
         <sub>ABC</sub> varchar(50)
                     _{A}^{B}_{C} char(1) _{A}^{B}_{C} varchar(50)
                                                                               <sub>ABC</sub> varchar(50)
                                                      <sub>ABC</sub> varchar(50)
                                                                    ABC char(1)
                                            # int
         陈医生
                     男
                               内科专家
                                                      李医生
                                                                     女
                                                                                外科专家
       2 王医生
                      男
                               儿科专家
```

2 赵医生

女

眼科专家

5. 自定义函数

男

3 孙医生

1) 特定用户的投诉次数(complaint_count)

心脏科专家

-- 投诉次数存储函数,输入一个用户 ID,可以返回他的投诉次数 delimiter // CREATE FUNCTION fun count com(user ID INT) RETURNS INT NO SQL **BEGIN DECLARE** cout int; **SELECT COUNT(*) into** cout **FROM** complaint user cu **WHERE** cu.user id = user ID; **RETURN** cout; END; // delimiter; -- 案列: 统计处 user_id = 1 的用户投诉次数 SELECT fun count com(1) as 投诉次数; 数据 信息 投诉次数 2

6. 建立存储过程

1) 列出开放问题(list_open_questions)

-- 列出开放问题,包括所有未回答的公开问题及当前登录专家的私有问题,按时间排序 delimiter //

CALL list_open_questions(1);

question_id	user_id	question_time	question_name	question_type	question_content	choice_expert
4	4	2024-12-02 20:08:02	营养补充	私有	我正在考虑服用复合维生	1
1	1	2024-11-29 11:10:29	饮食建议	公开	请问有哪些食物可以帮助	(Null)
5	5	2024-11-29 11:10:29	心理健康	公开	最近工作压力大,如何有法	(Null)

2) 指定日期的就诊情况(date_treat)

-- 查询特定日期的就诊情况,包括用户的诊断报告以及医生的诊断结果 delimiter //

```
CREATE PROCEDURE date_treat(IN report_date VARCHAR(10))
REGIN
```

CALL date treat('2024-11-29');

-- 案列: 查询 2024-11-29 当天的就诊情况

ureport_id	username	report_time	detail	result	discribe	treat_time
1	张三	2024-11-29 11:01:58	轻微头痛,伴有低烧。	普通感冒	患者报告轻微头痛和低烧	2024-11-29 11:05:17
2	李四	2024-11-29 11:01:58	胃部不适,感觉恶心。	急性胃炎	患者胃部不适, 恶心, 建	2024-11-29 11:05:17
3	王五	2024-11-29 11:01:58	持续咳嗽,有绿色痰液。	支气管炎	患者持续咳嗽,有绿色痰	2024-11-29 11:05:17
5	王十一	2024-11-29 11:01:58	视力模糊,伴有间歇性失力	视力问题	视力模糊,建议进行全面	2024-11-29 11:05:17

3. 建立触发器 两个

1) 不允许 18 岁以下家庭用户注册

```
-- 不允许 18 岁以下家庭用户注册
delimiter //
CREATE TRIGGER limit age
BEFORE INSERT ON 'user'
FOR EACH ROW
BEGIN
  IF new.user age <18 THEN
    SIGNAL SQLSTATE '45000' SET message text ='不允许 18 岁以下家庭用户注册';
  END IF;
END;
//
delimiter;
123
             张三
                             19
                                       12345678901
                                                                          就诊吃药
123
            李四
                            22
                                                         1 花粉讨敏
                                                                          定期复查
                                     0 12345678901
123
                             28
                                     1 12345678901
                                                         1 无
                                                                          正在接受抗生素治疗
123
            孙七
                            31
                                     0 44455566677
                                                         1 青霉素过敏
123
             王十-
                             21
                                       1122334455
                                                         0 无
                                                                          正在接受物理治疗
123
             小明
                                     1 12345678912
                             12
                       1644 - 不允许18岁以下家庭用户注册
                               确定
```

2) 修改密码日志 (update_pwd_log)

-- 记录家庭用户修改密码的所有操作,在 user pwd 表格中记录用户的旧密码,新密码

```
CREATE TABLE user_pwd(
   id INT auto_increment PRIMARY KEY,
   user_id INT,
   `name` VARCHAR(20),
   old_pwd VARCHAR(128),
   new_pwd VARCHAR(128)
);

delimiter //
CREATE TRIGGER record_pwd
AFTER UPDATE ON `user`
FOR EACH ROW
BEGIN
```

```
IF new.user_password IS NOT NULL THEN

INSERT INTO user_pwd (user_id, `name`, old_pwd, new_pwd)

VALUES (old.user_id, old.username,old.user_password,new.user_password);

ELSE

SIGNAL SQLSTATE '45000' SET message_text = '修改密码失败';

END IF;

END;

//

delimiter;

-- 案列: 修改张三的密码 zhangsan

UPDATE `user` SET user_password = 'zhangsan' WHERE user_id = 1;

UPDATE `user` SET user_password = 'wangwu' WHERE user_id = 3;
```

id # int	user_id # int	name _{A^BC} varchar(20)	old_pwd _{A^BC} varchar(128)	new_pwd _{A^BC} varchar(128)
1	1	张三	123	zhangsan
4	3	王五	123	wangwu

7. 总结

本系统是一个综合性的家庭医疗信息系统,旨在通过远程健康监测和在线咨询服务,提供全面的居家医疗支持。系统主要涉及两类用户,即家庭用户和在线专家。

对于家庭用户,他们可以通过系统注册,并享受远程医疗咨询服务。家庭用户可以录入 个人信息、健康状况、投诉、症状以及在家收集的诊断测试结果等。每位家庭用户可以有一 个指定的在线专家,负责提供个性化的医疗咨询服务。家庭用户通过系统可以向在线专家提 问,并根据专业建议进行相应的医疗活动。

在线专家是系统中的专业人士,包括临床医生、专家、全科医生、治疗师、保健顾问,他们通过系统登录,浏览家庭用户的健康信息,提供关于症状的意见,回答用户提出的问题,并指导家庭用户进行在家可收集的诊断测试。在线专家可以提供诊断、治疗建议,并根据家庭用户的健康状态进行进一步的调查。

系统记录了家庭用户的个人信息、健康状况、投诉、症状、诊断、疾病、诊断测试结果、治疗建议以及健康问题等信息。这些信息都有详细的记录,包括时间戳、严重程度、文字描述等。同时,系统也支持家庭用户和在线专家之间的交流,家庭用户可以提问保健问题,而在线专家可以回答,并对所提出的问题进行分类(私人或公开)。

总体而言,本系统为用户提供便捷的家庭医疗服务,通过记录和分析用户的健康信息,系统为用户和专业医护人员提供了一个有效的沟通平台,以实现更全面、个性化的医疗服务。

在这段数据库构建之旅中,我深刻领悟到了数据库设计的艺术性和科学性。将 E-R 图 巧妙地转化为关系模式,我掌握了将现实世界的复杂性凝练成数据库中的结构化设计,这不 仅提升了数据的组织效率,更为数据的检索与维护奠定了坚实的基石。在追求关系模式达到 第三范式(3NF)的征途中,我深刻体会到了数据规范化的深远意义,它不仅消除了数据冗余,避免了不一致性的困扰,还简化了数据的更新与维护流程。

通过深入学习和运用 PL/SQL,我得以实现数据完整性的自动化检查和业务逻辑的复杂处理,这不仅提升了数据处理的精确度,也极大地增强了系统的响应速度和适应性。从基础的 SQL 语言到高级的 PL/SQL,每一步的学习都让我对数据库的深邃和宽广有了更加深刻的理解。这些宝贵的经验无疑将对我的职业生涯产生深远的影响,同时,也点燃了我对计算机科学更深层次的热爱和探索欲望。