**Personal Health Database Project**

**by**

**Group 1**

SECTION 1: Overview

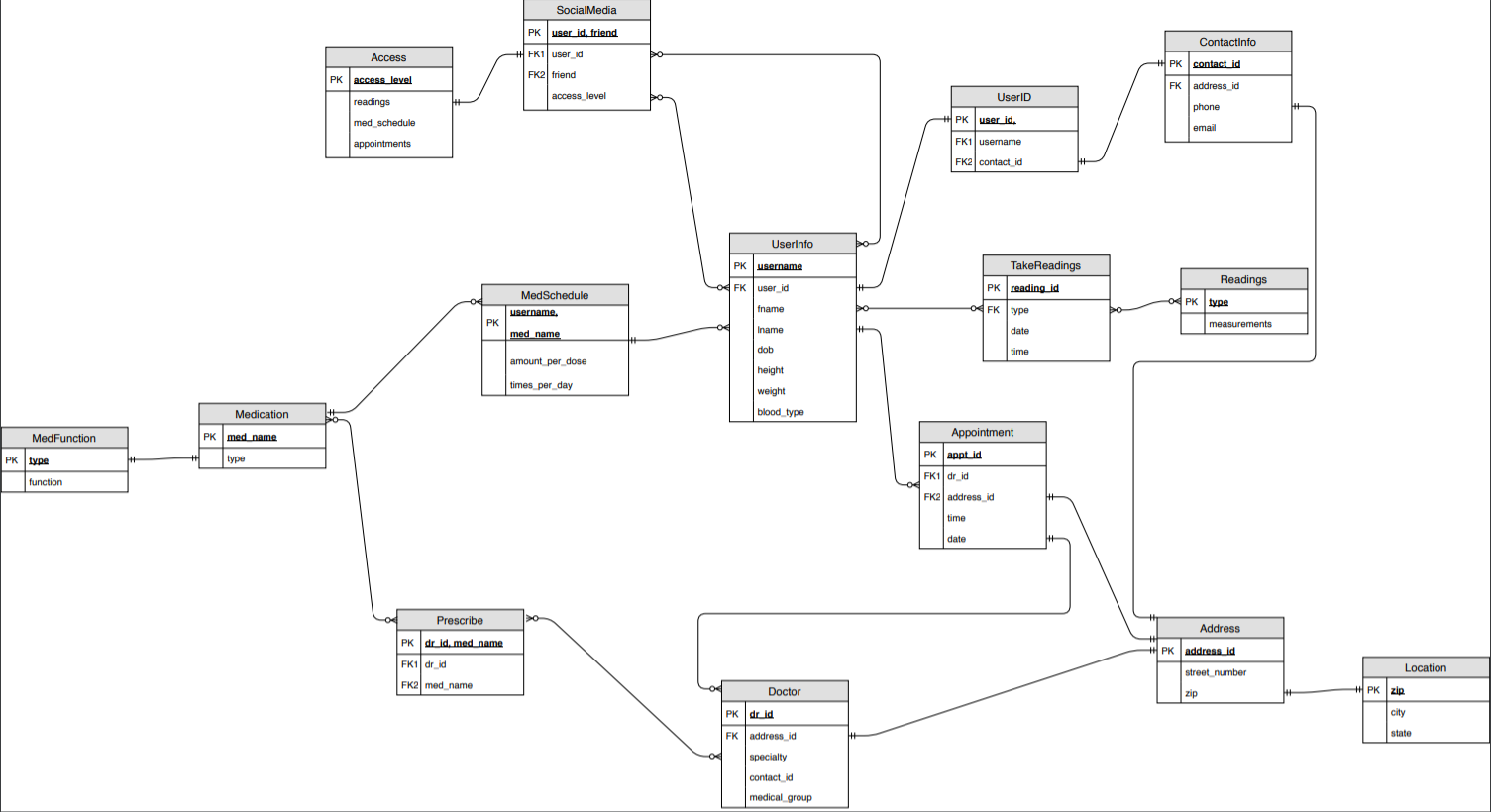
We have created a database to record and track personal health information. The database will also keep track of friends of the user as well as doctors that the user is seeing. This backend database can be thought of as the logical foundation for a personal health app. The purpose of this app is to include the power of the community to encourage healthy habits such as exercise, taking medication, and making and keeping doctors’ appointments. While the use of the app does encourage accountability with other people, it would also enable the user to limit the access of each “friend” in order to maintain privacy.

The specifics of our application’s functionality requires that the application must:

* Contain information for each user regarding contact information and health statistics such as height, weight, and blood type.
* Keep track of all medications that the user is currently taking. That information will include the type and quantity of medicine.
* Record and update levels of various measures of health status such as blood sugar levels, blood pressure, and/or hormone levels.
* Allow users to modify which levels are being recorded by the app.
* Maintain information regarding doctors’ appointments including location, date, and which doctor the appointment is with.
* Allow the user to select which types of information is accessible to each friend in the user’s friend list.
* Allow doctors to see medications that a user is taking in order to prescribe medication that isn’t harmful when mixing with other medications and to track the benefits of certain quantities of meds.
* Allow doctors to see a user’s health stats readings in order to monitor how the patient is responding to treatment.

The database that we have created facilitates the application’s functions by storing data in multiple ways. User information is stored as a collection of personal attributes such as height, weight, and blood type. For medications, the database stores the names of the medications that the user is taking, as well as the type of medicine and its function for the user. An example of the function for a user is if the type of medication is an antifungal perhaps the function for the user is for athlete’s foot. The users’ health measurements are recorded in the database by saving individual readings for each type and then adding a date and time stamp so that those readings can be evaluated over time. Also, each type of reading that is recorded is created by the user by updating the table and creating a new row. Doctors’ appointments can be stored and updated in the database by recording the doctor’s name, date, time, and location as well as some contact information for the doctor. The friends of a user, who are themselves also stored as users in the social media aspect of the database, are given an “access level” by the user which must be stored as non-null. The “access level” determines the types of information that is accessible. A view will be created that combines information from the doctor’s table with the medication table which will allow for a doctor to be notified of a user’s medication. Another view combines the user’s readings table with the doctor’s information table in order to keep the doctor up to date on the patient’s health.

SECTION 2: Comprehensive UML Diagram of the Database



SECTION 3: Entity and Attribute Lists

|  |  |  |  |
| --- | --- | --- | --- |
| **Relations** | Primary Key | Alternate Key | Foreign Key |
| UserID | user\_id |  | *username*, *contact\_id* |
| ContactInfo | contact\_id |  | *address\_id* |
| UserInfo | username |  | *user\_id* |
| Address | address\_id |  |  |
| Location | zip |  |  |
| SocialMedia | user\_id | friend | *user\_id*, *friend* |
| Access | access\_level |  |  |
| Readings | type |  |  |
| TakeReadings | reading\_id |  | *type* |
| Doctor | dr\_id |  | *address\_id* |
| Appointment | appt\_id |  | *address\_id*, *dr\_id* |
| Medication | med\_name |  |  |
| MedFunction | type |  |  |
| MedSchedule | username | med\_name | *username*, *med\_name* |
| Prescribe | dr\_id | med\_name |  |

SECTION 4: Normalization Forms

**1st Normal Form**:

Patient (user\_id, username, email, dob, fname, lname, height, weight, blood\_type, street, zip, city, state)

Friend (user\_id, username, email, dob, fname, lname)

SocialMedia (user\_id, access\_level, *friend)*

Readings (type, measurements)

TakeReadings (reading\_id, date, time, *type*)

Doctor (dr\_id, specialty, phone, email, medical\_group, street, zip, city, state)

Appointment (appt\_id, time, date, street, zip, city, state, *dr\_id*)

Medication (med\_name, type, function)

MedSchedule (username, med\_name, amount\_per\_dose, times\_per\_day)

Prescribe (dr\_id, med\_name)

**2nd Normal Form**:

Patient (user\_id, username, email, phone, dob, fname, lname, height, weight, blood\_type, *address\_id*)

Friend (user\_id, username, email, phone, dob, fname, lname)

SocialMedia (user\_id, access\_level, *friend*)

Readings (type, measurements)

TakeReadings (reading\_id, date, time, *type*)

Doctor (dr\_id, specialty, phone, email, medical\_group, *address\_id*)

Appointment (appt\_id, time, date, *address\_id*, *dr\_id*)

Address(address\_id, street\_number, zip)

Location (zip, city, state)

Medication (med\_name, type, function)

MedSchedule (username, med\_name, amount\_per\_dose, times\_per\_day)

Prescribe (dr\_id, med\_name)

**3rd Normal Form**:

UserID (user\_id, *username*, *contact\_id*)

ContactInfo (contact\_id, phone, email, address\_id)

UserInfo (username, fname, lname, dob, height, weight, blood\_type, *user\_id*)

Address(address\_id, street\_number, zip)

Location (zip, city, state)

SocialMedia (*user\_id*, *friend*, *access\_level*) /\* friend represents the user\_id of a connected user through social media \*/

Access (access\_level, readings, med\_schedule, appointments) /\* boolean values for readings, med\_schedule, and appointments allows a friend to be notified or to see a users attribute \*/

Readings (type, measurements)

TakeReadings (reading\_id, date, time, *type*)

Doctor (dr\_id, specialty, phone, email, medical\_group, *address\_id*)

Appointment (appt\_id, time, date, *address\_id*, *dr\_id*)

Medication (med\_name, type)

MedFunction (type, function)

MedSchedule (username, med\_name, amount\_per\_dose, times\_per\_day)

Prescribe (dr\_id, med\_name)

**BCNF**:

UserID (user\_id, *username*, *contact\_id*)

ContactInfo (contact\_id, phone, email, *address\_id*)

UserInfo (username, fname, lname, dob, height, weight, blood\_type, *user\_id*)

Address(address\_id, street\_number, zip)

Location (zip, city, state)

SocialMedia (*user\_id*, *friend*, access\_level) /\* friend represents the user\_id of a connected user through social media \*/

Access (access\_level, readings, med\_schedule, appointments) /\* boolean values for readings, med\_schedule, and appointments allows a friend to be notified or to see a users attribute \*/

Readings (type, measurements)

TakeReadings (reading\_id, date, time, *type*)

Doctor (dr\_id, specialty, contact\_id, medical\_group, *address\_id*)

Appointment (appt\_id, time, date, *address\_id*, *dr\_id*)

Medication (med\_name, type)

MedFunction (type, function)

MedSchedule (*username*, *med\_name*, amount\_per\_dose, times\_per\_day)

Prescribe (dr\_id, med\_name)

SECTION 4: Functional Dependencies

Patient

username->dob, fname, lname, email, height, weight, blood\_type, street, zip

Location

zip->city, state

Friend

username->dob, fname, lname, access\_level

Readings

type->measurements

TakeReadings

reading\_id->date, time,

Doctor

dr\_id->specialty, phone, email, medical\_group, hospital\_street, hospital\_zip

Appointment

appt\_id->time, date, hospital\_street, hospital\_zip, dr\_id

Medication

med\_name->type, function

MedSchedule

(username, med\_name)->amount\_per\_dose, times\_per\_day

Prescribe

(dr\_id, med\_name)

**Transitive:**

UserID

user\_id->username, contact\_id

Contact

contact\_id->phone, email, address\_id

IndividualInfo

Username->fname, lname, dob, height, weight, blood\_type

Medication

med\_name->type

Med\_Functionality

med\_type->med\_functionality

SECTION 5: Comprehensive Tables, Inserts, Deletes, Updates

**CREATING TABLES**

create table userid (

user\_id number(4) not null

constraint userid\_user\_id\_PK PRIMARY KEY,

username varchar(25) not null,

contact\_id number(4) not null

);

create table contact\_info (

contact\_id number(4) not null

constraint contactinfo\_contact\_id\_PK PRIMARY KEY,

phone varchar(20) not null,

email varchar(50) not null,

address\_id number(4)

);

create table user\_info (

username varchar(50) not null

constraint userinfo\_username\_PK PRIMARY KEY,

fname varchar(25) not null,

lname varchar(35) not null,

dob DATE not null,

height varchar(6) not null,

weight number(3) not null,

blood\_type varchar(3) not null,

user\_id number(4) not null

);

create table address (

address\_id number(4) not null

constraint address\_address\_id\_PK PRIMARY KEY,

street\_number varchar(30) not null,

zip number(5) not null

);

create table location (

zip number(5) not null

constraint location\_zip\_PK PRIMARY KEY,

city varchar(20) not null,

state varchar(2) not null

);

create table social\_media (

user\_id number(4) not null

constraint socialmedia\_user\_id\_PK PRIMARY KEY,

friend number(4),

access\_level number(1) not null

);

create table patient\_access (

access\_level number(1) not null

constraint access\_access\_level\_PK PRIMARY KEY,

readings number(1) not null,

med\_schedule number(1) not null,

appointments number(1) not null

);

create table readings (

reading\_type varchar(30) not null

constraint readings\_reading\_types\_PK PRIMARY KEY,

measurements varchar(20) not null

);

create table take\_readings (

reading\_id number(4) not null

constraint take\_readings\_reading\_id\_PK PRIMARY KEY,

reading\_date date not null,

reading\_time varchar(20) not null,

reading\_type varchar(30) not null

);

create table doctor (

dr\_id number(4) not null

constraint doctor\_dr\_id\_PK PRIMARY KEY,

specialty varchar(25) not null,

phone varchar(20) not null,

email varchar(50) not null,

medical\_group varchar(25) not null,

address\_id number(4) not null

);

create table appointment (

appt\_id number(4) not null

constraint appointment\_appt\_id\_PK PRIMARY KEY,

appt\_time varchar(20) not null,

appt\_date date not null,

address\_id varchar(4),

dr\_id number(4) not null

);

create table medication (

med\_name varchar(30) not null

constraint medication\_med\_name\_PK PRIMARY KEY,

med\_type varchar(30) not null

);

create table med\_function (

med\_type varchar(30) not null

constraint med\_function\_med\_type\_PK PRIMARY KEY,

med\_function varchar(50) not null

);

create table med\_schedule (

username varchar(25) not null

constraint med\_schedule\_username\_PK PRIMARY KEY,

med\_name varchar(30) not null,

amount\_per\_dose number(4) not null,

times\_per\_day number(3) not null

);

create table prescribe (

dr\_id number(4) not null

constraint prescribe\_dr\_id\_PK PRIMARY KEY,

med\_name varchar(30) not null

);

**ADDING FOREIGN KEYS**

alter table userid

add constraint userid\_username\_FK FOREIGN KEY(username) references user\_info(username);

alter table userid

add constraint userid\_contact\_id\_FK FOREIGN KEY (contact\_id) references contact\_info(contact\_id);

alter table contact\_info

add constraint contact\_info\_address\_id\_FK FOREIGN KEY (address\_id) references address(address\_id);

alter table user\_info

add constraint user\_info\_user\_id\_PK FOREIGN KEY (user\_id) references social\_media(user\_id);

alter table address

add constraint address\_zip\_FK FOREIGN KEY (zip) references locations(zip);

alter table social\_media

add constraint social\_media\_user\_id\_FK FOREIGN KEY (user\_id) references userid(user\_id);

alter table social\_media

add constraint social\_media\_friend\_FK FOREIGN KEY (friend) references userid(user\_id);

alter table take\_readings

add constraint take\_readings\_reading\_type\_FK FOREIGN KEY (reading\_type) references readings(reading\_type);

alter table doctor

add constraint doctor\_address\_id\_FK FOREIGN KEY (address\_id) references address(address\_id);

alter table appointment

add constraint appointment\_address\_id\_FK FOREIGN KEY (address\_id) references address(address\_id);

alter table appointment

add constraint appointment\_dr\_id\_FK FOREIGN KEY (dr\_id) references doctor(dr\_id);

alter table med\_schedule

add constraint med\_schedule\_username\_FK FOREIGN KEY (username) references user\_info(username);

alter table med\_schedule

add constraint med\_schedule\_med\_name\_FK FOREIGN KEY (med\_name) references medication(med\_name);

**INSERTION OF USER DATA**

**USER ID TABLE**

INSERT INTO userid(user\_id,username,contact\_id) VALUES (1,'SK',1);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (2,'DL',2);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (3,'PF',3);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (4,'JT',4);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (5,'EA',5);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (6,'WG',6);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (7,'JW',7);

INSERT INTO userid(user\_id,username,contact\_id) VALUES (8,'PV',8);

**CONTACT INFO TABLE**

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (1,'909-000-001','steven@gmail.com',1);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (2,'909-000-002','diana@gmail.com',2);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (3,'909-000-003','pat@gmail.com',3);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (4,'909-000-004','jonathan@gmail.com',4);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (5,'909-000-005','ellen@gmail.com',5);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (6,'909-000-006','william@gmail.com',6);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (7,'909-000-007','jennifer@gmail.com',7);

INSERT INTO contact\_info(contact\_id,phone,email,address\_id) VALUES (8,'909-000-008','peter@gmail.com',8);

**USER INFORMATION TABLE**

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('SK','Steven','King','01-JAN-1980','5''6''''',160,'A',1);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('DL','Diana','Lorentz','02-JAN-1980','5''7''''',157,'B',2);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('PF','Pat','Fay','03-JAN-1980','5''5''''',154,'AB',3);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('JT','Jonathan','Taylor','04-JAN-1980','5''6''''',170,'O',4);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('EA','Ellen','Abel','05-JAN-1980','5''7''''',165,'A',5);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('WG','William','Gietz','06-JAN-1980','6''0''''',175,'B',6);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('JW','Jennifer','Whalen','07-JAN-1980','5''9''''',171,'AB',7);

INSERT INTO user\_info(username,fname,lname,dob,height,weight,blood\_type,user\_id) VALUES ('PV','Peter','Vargas','08-JAN-1980','5''8''''',169,'O',8);

**ADDRESS TABLE**

INSERT INTO address(address\_id,street\_number,zip) VALUES (1, '731 Fondren', 75001);

INSERT INTO address(address\_id,street\_number,zip) VALUES (2, '638 Voss', 75002);

INSERT INTO address(address\_id,street\_number,zip) VALUES (3, '3321 Castle', 75003);

INSERT INTO address(address\_id,street\_number,zip) VALUES (4, '561 Rice', 75004);

INSERT INTO address(address\_id,street\_number,zip) VALUES (5, '890 Stone', 75005);

INSERT INTO address(address\_id,street\_number,zip) VALUES (6, '450 Berry', 75006);

INSERT INTO address(address\_id,street\_number,zip) VALUES (7, '980 Fire Oak', 75007);

INSERT INTO address(address\_id,street\_number,zip) VALUES (8, '975 Dallas', 75008);

INSERT INTO address(address\_id,street\_number,zip) VALUES (9, '9 Dallas', 75008);

INSERT INTO address(address\_id,street\_number,zip) VALUES (10, '7 Fondren', 75001);

INSERT INTO address(address\_id,street\_number,zip) VALUES (11, '3 Castle', 75003);

INSERT INTO address(address\_id,street\_number,zip) VALUES (12, '4 Berry', 75006);

INSERT INTO address(address\_id,street\_number,zip) VALUES (13, '5 Rice', 75004);

INSERT INTO address(address\_id,street\_number,zip) VALUES (14, '99 Dallas', 75008);

INSERT INTO address(address\_id,street\_number,zip) VALUES (15, '77 Fondren', 75001);

INSERT INTO address(address\_id,street\_number,zip) VALUES (16, '33 Castle', 75003);

INSERT INTO address(address\_id,street\_number,zip) VALUES (17, '44 Berry', 75006);

INSERT INTO address(address\_id,street\_number,zip) VALUES (18, '55 Rice', 75004);

**LOCATIONS TABLE**

INSERT INTO location(zip,city,state) VALUES (95001,'Houston','TX');

INSERT INTO location(zip,city,state) VALUES (95002,'Dallas','TX');

INSERT INTO location(zip,city,state) VALUES (95003,'Spring','TX');

INSERT INTO location(zip,city,state) VALUES (95004,'Fontana','CA');

INSERT INTO location(zip,city,state) VALUES (95005,'Los Angeles','CA');

INSERT INTO location(zip,city,state) VALUES (95006,'Bellaire','TX');

INSERT INTO location(zip,city,state) VALUES (95007,'Humble','TX');

INSERT INTO location(zip,city,state) VALUES (95008,'Seattle','WA');

**SOCIAL MEDIA TABLE**

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (1,2,0);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (2,3,1);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (3,4,2);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (4,5,3);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (5,6,4);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (6,7,5);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (7,8,6);

INSERT INTO social\_media(user\_id,friend,access\_level) VALUES (8,1,7);

**PATIENT ACCESS TABLE**

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (0,0,0,0);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (1,0,0,1);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (2,0,1,0);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (3,0,1,1);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (4,1,0,0);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (5,1,0,1);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (6,1,1,0);

INSERT INTO patient\_access(access\_level,readings,med\_schedule,appointments) VALUES (7,1,1,1);

**READINGS TABLE**

INSERT INTO readings(reading\_type,measurements) VALUES ('blood glucose','60 mg/dL');

INSERT INTO readings(reading\_type,measurements) VALUES ('systolic blood pressure','110 mmHg');

INSERT INTO readings(reading\_type,measurements) VALUES ('diastolic blood pressure','70 mmHg');

INSERT INTO readings(reading\_type,measurements) VALUES ('heart rate','80 bpm');

INSERT INTO readings(reading\_type,measurements) VALUES ('hemoglobin','15 gm/dL');

**TAKE READINGS TABLE**

INSERT INTO take\_readings(reading\_id,reading\_date,reading\_time,reading\_type) VALUES (1,'01-MAR-2020','08:20:00','blood glucose');

INSERT INTO take\_readings(reading\_id,reading\_date,reading\_time,reading\_type) VALUES (2,'02-MAR-2020','09:30:00','systolic blood pressure');

INSERT INTO take\_readings(reading\_id,reading\_date,reading\_time,reading\_type) VALUES (3,'03-MAR-2020','10:40:30','diastolic blood pressure');

INSERT INTO take\_readings(reading\_id,reading\_date,reading\_time,reading\_type) VALUES (4,'04-MAR-2020','12:10:50','heart rate');

INSERT INTO take\_readings(reading\_id,reading\_date,reading\_time,reading\_type) VALUES (5,'05-MAR-2020','20:10:20','hemoglobin');

INSERT INTO doctor(dr\_id,specialty,phone,email,medical\_group,address\_id) VALUES (1,'Allergist','909-100-001','dr1@gmail.com','Group1',9);

INSERT INTO doctor(dr\_id,specialty,phone,email,medical\_group,address\_id) VALUES (2,'Dermatologist','909-100-002','dr2@gmail.com','Group2',10);

INSERT INTO doctor(dr\_id,specialty,phone,email,medical\_group,address\_id) VALUES (3,'Ophthalmologist','909-100-003','dr3@gmail.com','Group1',11);

INSERT INTO doctor(dr\_id,specialty,phone,email,medical\_group,address\_id) VALUES (4,'Cardiologist','909-100-004','dr4@gmail.com','Group3',12);

INSERT INTO doctor(dr\_id,specialty,phone,email,medical\_group,address\_id) VALUES (5,'Gastroenterologist','909-100-005','dr5@gmail.com','Group2',13);

**APPOINTMENTS TABLE**

INSERT INTO appointment(appt\_id,appt\_time,appt\_date,address\_id,dr\_id) VALUES (1,'10:20:00','10-MAR-2020',14,1);

INSERT INTO appointment(appt\_id,appt\_time,appt\_date,address\_id,dr\_id) VALUES (2,'12:20:00','11-MAR-2020',15,2);

INSERT INTO appointment(appt\_id,appt\_time,appt\_date,address\_id,dr\_id) VALUES (3,'14:20:00','12-MAR-2020',16,3);

INSERT INTO appointment(appt\_id,appt\_time,appt\_date,address\_id,dr\_id) VALUES (4,'16:20:00','13-MAR-2020',17,4);

INSERT INTO appointment(appt\_id,appt\_time,appt\_date,address\_id,dr\_id) VALUES (5,'18:20:00','14-MAR-2020',18,5);

**MEDICATIONS TABLE**

INSERT INTO medication(med\_name,med\_type) VALUES ('Tylenol','Pain Reliever');

INSERT INTO medication(med\_name,med\_type) VALUES ('Advil','Pain Reliever');

INSERT INTO medication(med\_name,med\_type) VALUES ('TopCare','Immune Support');

INSERT INTO medication(med\_name,med\_type) VALUES ('Emergen-C','Immune Support');

INSERT INTO med\_function(med\_type, med\_function) VALUES ('Pain Reliever','Temporarily relieves minor ache and pains');

INSERT INTO med\_function(med\_type, med\_function) VALUES ('Immune Support','Help support your immune system');

**MEDICATION SCHEDULE TABLE**

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('SK','Tylenol',2,3);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('DL','Tylenol',3,3);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('PF','Advil',2,2);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('JT','Advil',2,3);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('EA','TopCare',1,2);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('WG','TopCare',1,2);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('JW','Emergen-C',1,3);

INSERT INTO med\_schedule(username,med\_name,amount\_per\_dose,times\_per\_day) VALUES ('PV','Emergen-C',1,1);

**PRESCRIBE TABLE**

INSERT INTO prescribe(dr\_id,med\_name) VALUES (1,'Tylenol');

INSERT INTO prescribe(dr\_id,med\_name) VALUES (2,'Advil');

INSERT INTO prescribe(dr\_id,med\_name) VALUES (3,'TopCare');

INSERT INTO prescribe(dr\_id,med\_name) VALUES (4,'Emergen-C');

INSERT INTO prescribe(dr\_id,med\_name) VALUES (5,'Advil');

**QUERIES**

**CREATING VIEWS**

/\* Create a view to show the detailed information of doctors. \*/

CREATE VIEW

Dr\_info

AS SELECT

d.dr\_id,

d.specialty,

d.phone,

d.email,

d.medical\_group,

d.address\_id,

a.street\_number,

a.zip,

l.city,

l.state

FROM

doctor d,

address a,

location l

WHERE

d.address\_id = a.address\_id

and a.zip = l.zip;

/\* Create a view to show the detailed information of users. \*/

CREATE VIEW

User\_med\_schedule

AS SELECT

ms.username,

ui.fname,

ui.lname,

c.phone,

c.email,

ui.dob,

ui.height,

ui.weight,

ui.blood\_type,

ms.med\_name,

ms.amount\_per\_dose,

ms.times\_per\_day

FROM

med\_schedule ms,

userid u,

user\_info ui,

contact\_info c

WHERE

u.username = ui.username

and ms.username = u.username

and u.contact\_id = c.contact\_id;

/\* Show the amount of medication per day for users whose weight is greater or equal than 165 order by their name. \*/

SELECT

fname,

lname,

weight,

sum(amount\_per\_dose \* times\_per\_day) AS Amount\_per\_day

FROM

User\_med\_schedule

GROUP BY

fname,

lname,

weight

HAVING

weight >= 165

ORDER BY

fname,

lname;

/\* Show the amount of group numbers in each group where the size is larger than 1 order by medical\_group name. \*/

SELECT

medical\_group,

count(medical\_group) AS Number\_of\_group\_member

FROM

Dr\_info

GROUP BY

medical\_group

HAVING

count(medical\_group) > 1

ORDER BY

medical\_group;

/\* Show the information in table prescribe join with table doctor, table medication and table med\_function. \*/

SELECT

p.dr\_id,

d.specialty,

d.phone,

d.email,

d.medical\_group,

p.med\_name,

m.med\_type,

mf.med\_function

FROM

prescribe p

left join

doctor d

on p.dr\_id = d.dr\_id

left join

medication m

on p.med\_name = m.med\_name

left join

med\_function mf

on m.med\_type = mf.med\_type;

/\* Show the access level of every user and their friends by join the table social\_media, table userid and table patient\_access. And the result means have access if 1 and no access if 0 \*/

SELECT

sm.user\_id,

u.username,

sm.friend AS friend\_id,

sm.access\_level,

a.readings,

a.med\_schedule,

a.appointments

FROM

social\_media sm

left join

userid u

on sm.user\_id = u.user\_id

left join

patient\_access a

on sm.access\_level = a.access\_level;

/\* Count the number of appointments where the address zip of appointment is larger than 95003. \*/

SELECT

COUNT(\*) AS Number\_of\_appointments

FROM

appointment a,

address ad,

location l

WHERE

a.address\_id = ad.address\_id

and ad.zip = l.zip

and ad.zip > 95003;

/\* Find how many medications named ‘Tylenol’ are taken by all users per day. \*/

SELECT

med\_name,

SUM(amount\_per\_dose \* times\_per\_day) AS Total\_med\_type\_taken

FROM

med\_schedule

WHERE

med\_name = 'Tylenol'

GROUP BY

med\_name;

/\* Update the user info of JW, change blood\_type to O and change date of birth to 07-FEB-1980.\*/

UPDATE

user\_info

SET

blood\_type = 'O',

dob = '07-FEB-1980'

WHERE

username = 'JW'

and fname = 'Jennifer'

and lname = 'Whalen';

/\* Update the medication schedule of user ‘JT’ and change med\_name to ‘Tylenol’ and times\_per\_day to 2. \*/

UPDATE

med\_schedule

SET

med\_name = 'Tylenol',

times\_per\_day = 2

WHERE

username = 'JT';

/\* Insert into table address and location new data. \*/

INSERT INTO address(address\_id,street\_number,zip) VALUES (19, '5500 University', 92407);

INSERT INTO location(zip, city, state) VALUES (92407, 'San Bernardino', 'CA');

**DELETION OF ALL TABLES**

DROP TABLE userid;

DROP TABLE contact\_info;

DROP TABLE user\_info;

DROP TABLE address;

DROP TABLE location;

DROP TABLE social\_media;

DROP TABLE patient\_access;

DROP TABLE readings;

DROP TABLE take\_readings;

DROP TABLE doctor;

DROP TABLE appointment;

DROP TABLE medication;

DROP TABLE med\_function;

DROP TABLE med\_schedule;

DROP TABLE prescribe;