Database Project

Christina Ye
hy2084@nyu.edu

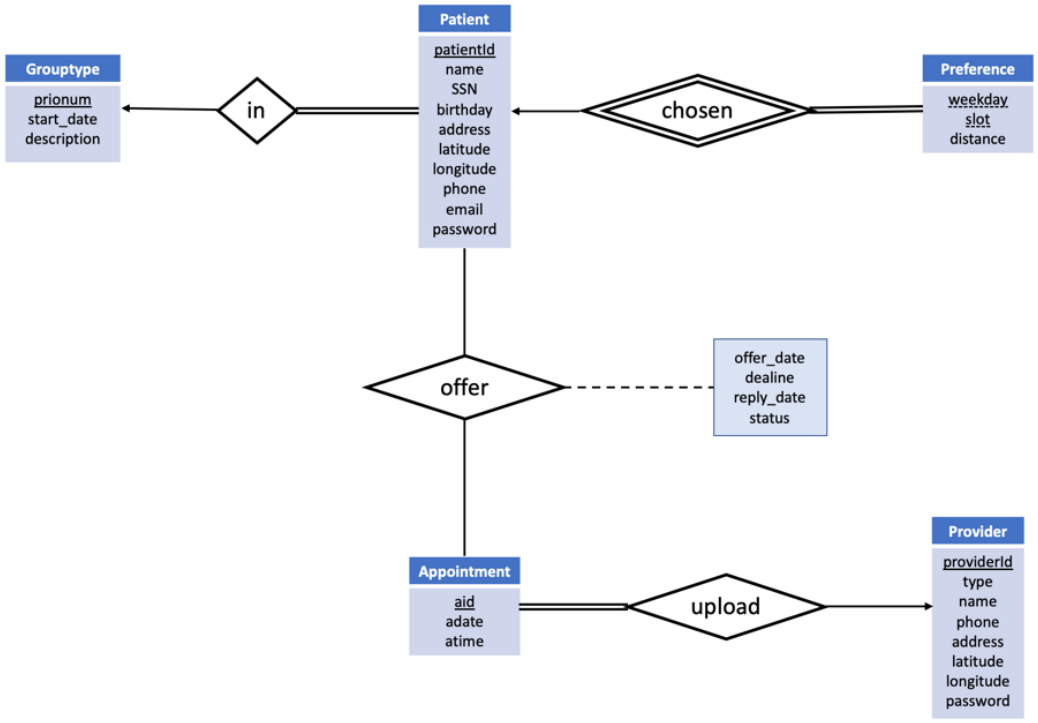
Part 1 Relational Database Design

Introduction

In order to create a web-based system for registering covid-19 vaccination information and realize the functionalities such as registering patients, adding available appointments and offering appointments to patients, we first built up the schema for the databases and created the corresponding tables, some sample data were inserted to test the functionality of the model by applying the queries.

(a) Relational Schema & E-R diagram

E-R diagram:



Relational schema:

(1) **Grouptype** (prionum,start_date,description)
Primary key: (prionum)

prionum: number between 1-4 that specify the group a patient belongs to
start_date: a date after which the patient in the group is eligible to get vaccination
description: the additional information for the group, such as age, career

(2) **Patient**
(patientId,name,SSN,birthday,address,latitude,longitude,phone,email,prionum,password)
Primary key: (patientId)
prionum references **Grouptype(prionum)**

patientId: it is unique and refers to the username of the patient in this system
name,SSN,birthday,address,phone,email,: patient's information entered when registering
latitude,longitude: the coordination of the address, will be used to calculate distances
prionum: the group which the patient belongs to, will be derived by calculation and other documents
password: created when the patient registered, will be used to authenticate the patient when login

(3) **Provider**(providerId,type,name,phone,address,latitude,longitude,password)
Primary key: (providerId)

providerId: it is unique and refers to the username of the provider in this system
type: the type of the provider, for example, it can be "hospital" or "pharmacy"
name,phone,address: provider's information entered when registering
latitude,longitude: the coordination of the address, will be used to calculate distances
password: created when the provider registered, will be used to authenticate the patient when login

(4) **Preference**(patientId,weekday,slot,distance)
Primary key: (patientId, weekday, slot)
patientId references **Patient(patientId)**

patientId: refer to the patient who enters this preference
weekday,slot: weekday will be numbers between 0-7, slot can be 8:00 or 12:00, which refers to time block 8:00am-12:00pm and 12:00pm-16:00pm
distance: the longest distance the patient can accept

(5) **Appointment**(aid,adate,atime,providerId)
Primary key: (aid)
providerId references **Provider(providerId)**

aid: the id of the appointment, will be automatically generated when inserting new appointment
adate, atime: the date and start time of the appointment
providerId: the provider who entered the appointment

(6) **Offer**(aid,patientId,offer_date,deadline,reply_date,status)
Primary key: (aid, patientId)
aid references **appointment(aid)**
patientId references **patient(patientId)**

aid,patientId,offer_date: means that the appointment with the aid was assigned to patient with patientId on offer_date
deadline: the latest date the offer can be accepted, if the offer was not replied after this date, it will be marked as "declined"
reply_date: the date the patient reply to this offer
status: can be "unconfirmed", "accepted", "declined, "cancelled", "missed" and "completed". The offer can be cancelled after it is being accepted; The offer can be marked as "missed" if patient did not show up and "completed" if the appointment is completed, in other words, the patient received the vaccination

Functionalities of the system:

(1) Manage account

The patients and providers can register accounts using different links, their account information and other information such as name, address will be stored in "patient" and "provider" tables.

(2) Add appointment & add preference

After a provider login to its own account, it can add appointments; while the patient can login to his/her account to add a slot preference. The new added appointment and preference will be stored in "appointment" and "preference" table

(3) Assigning appointment

The administrator can run algorithms to assign the available appointment to patients according to their preference and current availabilities, the assignment records will be stored in "offer" table

(4) Update offer

When the appointment is assigned, the status will be initialized to "unconfirmed" until the patient replies.
When a patient accept an appointment, the status of this "offer" change to "accepted";
When a patient decline an appointment, the status of this "offer" change to "declined";
When a patient did not respond within deadline, the status of this "offer" change to "declined";
When a patient cancel the appointment after the accepting, the status of this "offer" change to "cancelled";
When a patient did not show up, the status of this "offer" change to "missed";
When a patient received vaccination, the status of this "offer" change to "completed";

(b) Create database using MySQL

Create table statements:

```
DROP TABLE IF EXISTS offer;
DROP TABLE IF EXISTS appointment;
```

```
DROP TABLE IF EXISTS preference;
DROP TABLE IF EXISTS provider;
DROP TABLE IF EXISTS patient;
DROP TABLE IF EXISTS grouptype;


CREATE TABLE grouptype(
  prionum INTEGER primary key,
  startdate DATE,
  description VARCHAR(50)
);

CREATE TABLE patient(
        patientId VARCHAR(20) primary key,
        name VARCHAR(20),
        SSN VARCHAR(9),
        birthday DATE,
        address VARCHAR(50),
        latitude FLOAT,
        longitude FLOAT,
        phone VARCHAR(11),
        email varchar(30),
        prionum INTEGER,
        password VARCHAR(20),
        FOREIGN KEY (prionum) REFERENCES grouptype(prionum)
);

CREATE TABLE provider(
        providerId VARCHAR(20) primary key,
        providertype VARCHAR(10),
        name VARCHAR(50),
        phone VARCHAR(11),
        address VARCHAR(50),
        latitude FLOAT,
        longitude FLOAT,
        password VARCHAR(20)
);

CREATE TABLE preference(
        patientId VARCHAR(20),
        weekday INTEGER,
        slot TIME,
        distance INTEGER,
        PRIMARY KEY (PatientId, weekday, slot),
        FOREIGN KEY (PatientId) REFERENCES patient(PatientId)
);
```

```
CREATE TABLE appointment(
        aid INTEGER NOT NULL AUTO_INCREMENT,
        adate DATE,
        atime TIME,
        providerId VARCHAR(20),
        PRIMARY KEY (aid),
        FOREIGN KEY (providerId) REFERENCES provider(providerId)
);

CREATE TABLE offer(
        aid INTEGER,
        patientId VARCHAR(20),
        offerdate DATE,
        deadline DATE,
        replydate DATE,
        status VARCHAR(10),
        PRIMARY KEY (aid, patientId),
        FOREIGN KEY (aid) REFERENCES appointment(aid),
        FOREIGN KEY (patientId) REFERENCES patient(patientId)
);
```

The sample data are loaded through MySQL workbench.

Example data after the insertion:

select * from p.grouptype

prionum	startdate	description
1	2021-03-01	doctors and nurses
2	2021-03-15	60-year-olds-up
3	2021-04-01	10-year-olds-below
4	2021-04-15	other people

select * from p.patient

patientId	name	SSN	birthday	address	latitude	longitude	phone	email	prionum	password
bob0823	Bob	293019374	1983-08-23	824 Park Ave, Brooklyn, NY	40.6981	-73.9424	4982909274	bob0823@gmail.com	1	123123123
gabriel1105	Gabriel	629204862	1970-11-05	390 Gates Ave, Brooklyn	40.6861	-73.9509	5829902749	gabriel@gmail.com	4	19701105
jj999	Jack	628374018	1969-10-09	105 E 22nd St,New York, NY	40.7394	-73.9864	8207468290	jj999@wisc.edu	4	jjjj9999
jones90	Jones	829380921	1990-09-02	343 Gold Street, Brooklyn, NY	40.6937	-73.9829	2398928398	js@nyu.edu	1	password
mary2021	Mary	108374928	2021-02-03	22 E 119th St, New York, NY	40.8021	-73.9442	2019284038	mamaryry@gmail.com	3	password
michael6	Michael	820471928	1950-12-11	138-18 Northern Blvd, Flushing, NY	40.7107	-74.011	2890928392	michael50@gmail.com	2	test123
ronron0607	Ron	402948465	1959-06-07	163 Eastern Pkwy, Brooklyn, NY	40.6723	-73.9636	3294890289	ron0607@gmail.com	2	ron1959
rose1999	Rose	820174927	1999-03-02	1035 Washington Ave, Brooklyn, NY	40.6646	-73.9604	8019309210	roserose@gmail.com	4	rose19990302

select * from p.provider

providerId	providertype	name	phone	address	latitude	longitude	password
cvs1	pharmacy	CVS Pharmacy	2129825193	300 Park Ave S, New York, NY 10010	40.7398	-73.9869	cvspharmacy
flushinghospital	hospital	Flushing Hospital Medical Center	7186705000	4500 Parsons Blvd, Flushing, NY	40.7555	-73.8166	flushing66
nycmetropolitan	hospital	NYC Health + Hospitals/Metropolitan	2124236262	1901 1st Avenue, New York, NY 10029	40.7851	-73.9451	metro123
nycwoodhull	hospital	NYC Health + Hospitals/Woodhull	7189638000	760 Broadway, Brooklyn, NY 11206	40.6995	-73.9426	password
walgreen1	pharmacy	Walgreens Pharmacy	7184039112	379 Myrtle Ave, Brooklyn, NY 11205	40.6933	-73.971	123456
walgreen2	pharmacy	Walgreens Pharmacy	7183982074	1281 Fulton St, Brooklyn, NY 11216	40.6805	-73.9502	wgr222

select * from p.preference

patientId	weekday	slot	distance
bob0823	2	08:00:00	2
gabriel1105	3	12:00:00	2
gabriel1105	4	12:00:00	2
jj999	0	08:00:00	1
jj999	0	12:00:00	1
jj999	2	08:00:00	1
jj999	2	12:00:00	1
jones90	1	08:00:00	1
jones90	2	08:00:00	1
jones90	3	08:00:00	1
jones90	5	08:00:00	1
mary2021	1	12:00:00	3
mary2021	2	12:00:00	3
mary2021	5	12:00:00	7
mary2021	6	12:00:00	7
michael6	5	12:00:00	10
michael6	6	12:00:00	10
ronron0607	0	08:00:00	5
ronron0607	1	08:00:00	5
ronron0607	2	08:00:00	5
ronron0607	3	08:00:00	5
rose1999	4	12:00:00	6

select * from p.appointment

aid	adate	atime	providerId
1	2021-03-02	09:00:00	walgreen1
2	2021-03-02	14:00:00	walgreen1
3	2021-03-03	09:00:00	walgreen1
4	2021-03-03	14:00:00	walgreen1
5	2021-03-04	09:00:00	walgreen1
6	2021-03-04	14:00:00	walgreen1
7	2021-04-03	09:30:00	walgreen1
8	2021-04-03	13:30:00	walgreen1
9	2021-04-04	09:30:00	walgreen1
10	2021-04-04	13:30:00	walgreen1
11	2021-05-08	09:30:00	walgreen1
12	2021-03-01	10:00:00	nycwood…
13	2021-03-02	10:00:00	nycwood…
14	2021-03-07	15:00:00	nycwood…
15	2021-03-08	15:00:00	nycwood…
16	2021-03-15	10:00:00	nycwood…
17	2021-03-16	10:00:00	nycwood…
18	2021-04-05	15:00:00	nycwood…
19	2021-04-06	15:00:00	nycwood…
20	2021-04-12	10:00:00	nycwood…
21	2021-04-13	10:00:00	nycwood…
22	2021-05-03	10:00:00	nycwood…
23	2021-05-04	10:00:00	nycwood…
24	2021-05-12	15:00:00	nycwood…
25	2021-05-13	15:00:00	nycwood…
26	2021-04-15	15:00:00	walgreen2
27	2021-04-17	15:00:00	walgreen2
28	2021-04-19	15:00:00	walgreen2
29	2021-04-21	15:00:00	walgreen2
30	2021-04-23	15:00:00	walgreen2
31	2021-04-25	15:00:00	walgreen2

32	2021-04-27	15:00:00	walgreen2
33	2021-04-29	15:00:00	walgreen2
34	2021-03-22	10:00:00	cvs1
35	2021-03-22	14:00:00	cvs1
36	2021-03-24	10:00:00	cvs1
37	2021-03-24	14:00:00	cvs1
38	2021-03-27	10:00:00	cvs1
39	2021-03-27	14:00:00	cvs1
40	2021-03-02	13:00:00	nycmetr…
41	2021-03-03	13:00:00	nycmetr…
42	2021-03-04	13:00:00	nycmetr…
43	2021-03-05	13:00:00	nycmetr…
44	2021-03-06	13:00:00	nycmetr…
45	2021-03-07	13:00:00	nycmetr…
46	2021-03-06	14:00:00	flushingh…
47	2021-03-07	08:00:00	flushingh…
48	2021-03-14	14:00:00	flushingh…
49	2021-03-15	08:00:00	flushingh…
50	2021-03-20	14:00:00	flushingh…
51	2021-03-21	08:00:00	flushingh…
52	2021-03-28	14:00:00	flushingh…
53	2021-04-26	08:00:00	flushingh…
54	2021-05-01	14:00:00	flushingh…
55	2021-05-14	08:00:00	flushingh…
NULL	NULL	NULL	NULL

select * from p.offer

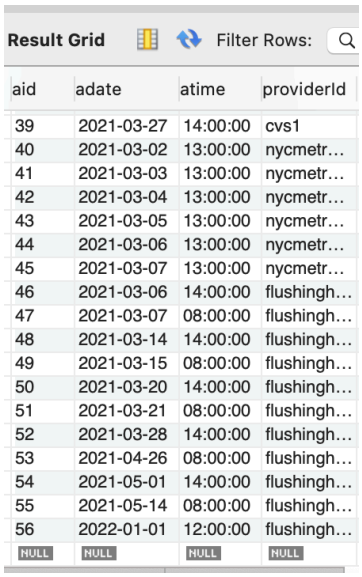
aid	patientId	offerdate	deadline	replydate	status
1	jones90	2021-02-20	2021-02-27	2021-02-24	missed
3	jones90	2021-02-20	2021-02-27	2021-02-24	missed
3	ronron0607	2021-02-20	2021-02-27	2021-02-21	cancelled
5	ronron0607	2021-02-20	2021-02-27	2021-02-21	cancelled
10	michael6	2021-02-20	2021-02-27	2020-02-24	completed
12	ronron0607	2021-02-20	2021-02-27	2021-02-21	cancelled
14	michael6	2021-02-20	2021-02-27	2021-02-24	declined
16	ronron0607	2021-02-20	2021-02-27	2021-02-21	declined
25	gabriel1105	2021-04-21	2021-04-28	2021-04-22	accepted
30	rose1999	2021-03-21	2021-03-28	2021-03-26	accepted
34	jj999	2021-02-21	2021-02-28	2021-02-26	completed
40	mary2021	2021-02-20	2021-02-27	2021-02-27	declined
41	mary2021	2021-02-20	2021-02-27	2021-02-25	completed
NULL	NULL	NULL	NULL	NULL	NULL

(c) SQL queries

(1)
INSERT INTO p.patient value("john98","John Smith","123456789","1999-01-01","1000 University Ave, New York, NY",40,-70,2222222222,"jacksmith@gmail.com",4,"123123")

patientId	name	SSN	birthday	address	latitude	longitude	phone	email	prionum	password
bob0823	Bob	293019374	1983-08-23	824 Park Ave, Brooklyn, NY	40.6981	-73.9424	4982909274	bob0823@gmail.com	1	123123123
gabriel1105	Gabriel	629204862	1970-11-05	390 Gates Ave, Brooklyn	40.6861	-73.9509	5829902749	gabriel@gmail.com	4	19701105
jj999	Jack	628374018	1969-10-09	105 E 22nd St,New York, NY	40.7394	-73.9864	8207468290	jj999@wisc.edu	4	jjjj9999
john98	John Smith	123456789	1999-01-01	1000 University Ave, New York, NY	40	-70	2222222222	jacksmith@gmail.com	4	123123
jones90	Jones	829380921	1990-09-02	343 Gold Street, Brooklyn, NY	40.6937	-73.9829	2398928398	js@nyu.edu	1	password
mary2021	Mary	108374928	2021-02-03	22 E 119th St, New York, NY	40.8021	-73.9442	2019284038	mamaryry@gmail.com	3	password
michael6	Michael	820471928	1950-12-11	138-18 Northern Blvd, Flushing, NY	40.7107	-74.011	2890928392	michael50@gmail.com	2	test123
ronron0607	Ron	402948465	1959-06-07	163 Eastern Pkwy, Brooklyn, NY	40.6723	-73.9636	3294890289	ron0607@gmail.com	2	ron1959
rose1999	Rose	820174927	1999-03-02	1035 Washington Ave, Brooklyn, NY	40.6646	-73.9604	8019309210	roserose@gmail.com	4	rose19990302

(2)
INSERT INTO p.appointment(adate,atime,providerId) value("2022-01-01","12:00","flushinghospital")



Result Grid Filter Rows:

aid	adate	atime	providerId
39	2021-03-27	14:00:00	cvs1
40	2021-03-02	13:00:00	nycmetr…
41	2021-03-03	13:00:00	nycmetr…
42	2021-03-04	13:00:00	nycmetr…
43	2021-03-05	13:00:00	nycmetr…
44	2021-03-06	13:00:00	nycmetr…
45	2021-03-07	13:00:00	nycmetr…
46	2021-03-06	14:00:00	flushingh…
47	2021-03-07	08:00:00	flushingh…
48	2021-03-14	14:00:00	flushingh…
49	2021-03-15	08:00:00	flushingh…
50	2021-03-20	14:00:00	flushingh…
51	2021-03-21	08:00:00	flushingh…
52	2021-03-28	14:00:00	flushingh…
53	2021-04-26	08:00:00	flushingh…
54	2021-05-01	14:00:00	flushingh…
55	2021-05-14	08:00:00	flushingh…
56	2022-01-01	12:00:00	flushingh…
NULL	NULL	NULL	NULL

he appointment is stored and it got an automatically generated aid “56”

(3)

```
with t1 as(
  select aid,adate,atime,providerId,slot,patientId,distance
  from test.preference,test.appointment
  where weekday(adate)=test.preference.weekday and
  TIMESTAMPDIFF(hour, slot, atime) >= 0 and
  TIMESTAMPDIFF(hour, slot, atime)<4 and
  adate > "2021-03-10" and
  patientId="michael6"),
t2 as (
  select aid, adate, atime, t1.providerId, t1.patientId, distance,
        111.111 * 0.62
        * DEGREES(ACOS(LEAST(1.0, COS(RADIANS(test.provider.latitude))
        * COS(RADIANS(test.patient.latitude))
        * COS(RADIANS(test.provider.longitude - test.patient.longitude))
        + SIN(RADIANS(test.provider.latitude))
        * SIN(RADIANS(test.patient.latitude))))) AS distance_in_miles
        from t1, test.provider, test.patient
        where test.provider.providerId=t1.providerId and
        test.patient.patientId=t1.patientId)
select patientId, providerId, aid, adate, atime, distance_in_miles from t2
where distance_in_miles <= distance and
```

aid not in (select aid from test.offer where status="accepted" or status="completed" or status="unconfirmed")
order by distance_in_miles

patientId	providerId	aid	adate	atime	distance_in_miles
michael6	cvs1	39	2021-03-27	14:00:00	2.3697863291609442
michael6	walgreen1	8	2021-04-03	13:30:00	2.409437885288791
michael6	walgreen2	27	2021-04-17	15:00:00	3.793889939396846
michael6	walgreen2	31	2021-04-25	15:00:00	3.793889939396846

The first table “t1” joins appointment with patient’s preference, it selects all of the appointment according to the weekday and time in the preference, and it also filtered out the date before “2021-03-10” assuming this is the current date and we want appointment after this date;
The second table “t2” joins t1 with patient and provider, it also calculate the distance between the patient and provider;
Finally we select the appointments that are within the distance preference and are not “unconfirmed”, “accepted” or “completed”, the final result is sort by patientId and distance.

(4)

```
with scheduled as (
  select distinct p.offer.patientId, prionum
  from p.offer,p.patient
  where status="accepted"
  and p.offer.patientId=p.patient.patientId),
received as (
  select distinct p.offer.patientId, prionum
  from p.offer,p.patient
  where status="completed" and
  p.offer.patientId=p.patient.patientId),
waiting as (
  select p.patient.patientId, prionum
  from p.patient
  where p.patient.patientId not in
  (select patientId from scheduled union select patientId from received)),
scheduled_num as (
  select prionum, count(*) as scheduled_count
  from scheduled group by prionum),
received_num as (
  select prionum, count(*) as received_count
  from received group by prionum),
waiting_num as (
  select prionum, count(*) as waiting_count
  from waiting group by prionum)
select p.grouptype.prionum,
  coalesce(scheduled_count, 0) as scheduled_number,
  coalesce(received_count, 0) as received_number,
  coalesce(waiting_count, 0) as wating_number
```

```
  from p.grouptype
  left join scheduled_num on p.grouptype.prionum=scheduled_num.prionum
  left join received_num on p.grouptype.prionum=received_num.prionum
  left join waiting_num on p.grouptype.prionum=waiting_num.prionum
```

prionum	scheduled_number	received_number	wating_number
1	0	0	2
2	0	1	1
3	0	1	0
4	2	1	0

First three tables “scheduled”, “received”, “waiting” are created to select the patients who got the appointments that are “accepted”, “completed” with their group number and the patient who have not accepted offer or completed vaccination with their group number
The following three tables “scheduled_num”, “received_num”, “waiting_num” count the number of patients in different priority groups
Finally we join grouptype with the numbers, the null values are substituted by “0”

(5)

```
select p.patient.prionum, p.patient.patientId, p.patient.name, startdate
from p.patient,p.grouptype
where p.patient.prionum=p.grouptype.prionum
```

prionum	patientId	name	startdate
1	bob0823	Bob	2021-03-01
1	jones90	Jones	2021-03-01
2	michael6	Michael	2021-03-15
2	ronron0607	Ron	2021-03-15
3	mary2021	Mary	2021-04-01
4	gabriel1105	Gabriel	2021-04-15
4	jj999	Jack	2021-04-15
4	rose1999	Rose	2021-04-15

The result is derived by join grouptype and patient on their group number.

(6)

```
with t1 as(
  select patientId, count(*) as missed_count
  from p.offer
  where status="missed"
  group by patientId
  having count(*) >= 2),
t2 as(
  select patientId, count(*) as cancelled_count
  from p.offer
  where status="cancelled"
  group by patientId
  having count(*) >= 3)
select p.patient.patientId, name
```

```
from p.patient
where p.patient.patientId in (select patientId from t1)
or p.patient.patientId in (select patientId from t2)
```

patientId	name
jones90	Jones
ronron0607	Ron

The first table t1 select the patientId and the times they missed accepted appointments, and those who have missed 2 or more were selected
The first table t1 select the patientId and the times they cancelled appointments, and those who have missed 3 or more were selected
Finally we select the patients whose id belongs to these two groups ("Jones" has missed 2 times and "Ron" has cancelled 3 times)

(7)

```
WITH t1 AS (
  select providerId, count(*) as ct
  from p.offer,p.appointment
  where status="completed" and
  p.offer.aid=p.appointment.aid
  group by providerId),
t2 as (select max(ct) as m from t1)
select p.provider.providerId, name, t1.ct
from p.provider, t1, t2
where p.provider.providerId=t1.providerId and t1.ct=m
```

providerId	name	ct
walgreen1	Walgreens Pharmacy	1
cvs1	CVS Pharmacy	1
nycmetropolitan	NYC Health + Hospitals/Metropolitan	1

Since when a patient is vaccinated, the corresponding offer will be marked as "completed", table t1 filter out these "completed" offer and gets the corresponding provider information, and calculate each provider's count
t2 selects the max number from t1, which is the "largest number of vaccination"
finally providers who has number of vaccination that is equal to the "largest number of vaccination" are selected

Part 2 Web Application Design

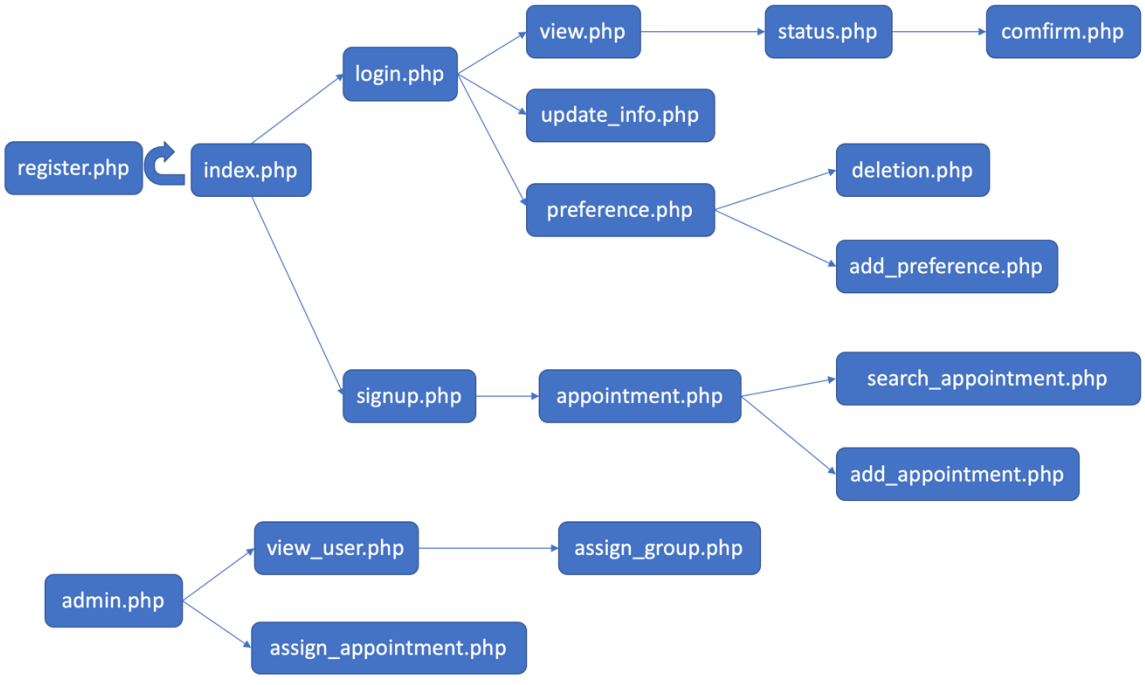
(a) Revision of the Relational Schema

The original relational schema from part 1 was already able to realize the function of the system. In order to show the effect of the date on the result, the appointment date and eligible date in the example data is updated so some groups are eligible to get the vaccination from today but some are still not. In addition, there are some constraints on some of the attribute, which will be realized by combining the web application:

1) When inserting into **appointment**, the **adate** must be at least 7 days after today
2) When inserting into or update **offer**, the **offerdate** must be at least 7 days before the appointment date; The **deadline** is exactly 7 days after the **offerdate**; The **replydate** must be no later than the **deadline** (otherwise the patient cannot change the status of this offer); If the **status** equal to "offer", it can be updated to "declined" or "accepted"; If the **status** equal to "declined", "cancelled" or "missed", it cannot be changed to other status; if the **status** equal to "accept", it can be updated to "cancelled" by patient, or "complete" or "missed" by provider.

(b) Webpage Design

The Overall Structure of the Application



Functionalities of the Webpage

Index

If the user (patient or provider) is not logged in, it will show the message of login/register/signup options. The registration uses GoogleMap API to get address's longitude and latitude and store them in the database.

You are currently not logged in.

Click login to login or register if you don't have an account yet.

Click Signup to login if you are a provider.

Patient Guide

For logged in patient, it will show options of viewing the offer, update their personal information, add or delete preferences or logout

Welcome Jack. You are logged in to the system.

Your Information:
UserID:jj999
Name:Jack
SSN:628374018
Birthday:1969-10-09
Address:105 E 22nd St,New York, NY
Phone:8207468290
E-Mail:huanyao0718@gmail.com

To view the your appointment offers, click here,

Update information: click here,

View or add preferences: click here,

Log out: click here.

Viewing the offer and update status

If the user choose to view his/her offer, a list of the appointments will be shown

Example of user "jj999"
SQL query:

```
select appointment.aid, provider.name, provider.phone, provider.address, 111.111 * 0.62
        * DEGREES(ACOS(LEAST(1.0, COS(RADIANS(provider.latitude))
        * COS(RADIANS(patient.latitude))
        * COS(RADIANS(provider.longitude - patient.longitude))
        + SIN(RADIANS(provider.latitude))
        * SIN(RADIANS(patient.latitude))))) AS distance_in_miles,
        adate, atime, offerdate, deadline, replydate, status from offer, appointment, provider,
patient where offer.aid=appointment.aid and appointment.providerId=provider.providerId and
offer.patientId=patient.patientId and patient.patientId= "jj999 "
```

Provider	Phone	Address	Distance	Appointment Date	Appointment Time	Date Offered	Acception Deadline	Your Reply Date	Status	Change Status
NYC Health + Hospitals/Woodhull	7189638000	760 Broadway, Brooklyn, NY 11206	3.5772117129601	2021-06-10	11:00:00	2021-05-18	2021-05-25		offer	change
CVS Pharmacy	2129825193	300 Park Ave S, New York, NY 10010	0.035024297210153	2021-07-22	10:00:00	2021-05-15	2021-05-22	2021-05-16	cancelled	change
CVS Pharmacy	2129825193	300 Park Ave S, New York, NY 10010	0.035024297210153	2021-07-22	14:00:00	2021-05-15	2021-05-22	2021-05-16	cancelled	change
CVS Pharmacy	2129825193	300 Park Ave S, New York, NY 10010	0.035024297210153	2021-07-24	10:00:00	2021-05-15	2021-05-22	2021-05-16	cancelled	change
CVS Pharmacy	2129825193	300 Park Ave S, New York, NY 10010	0.035024297210153	2021-07-24	14:00:00	2021-05-15	2021-05-22	2021-05-18	cancelled	change

Go back

Note that the user cannot update the appointment if the status is cancelled/declined/missed; If the status is “offer” the user can accept/decline the offer:

You can accept or decline this offer. Accept Decline

Go back to your appointments, click here

Example SQL query:
(1) Check if the appointment is valid first
select status from offer where patientId="jj999 " and aid=41 and CURDATE() <= deadline
(2) update with corresponding action if valid
update offer set status="accept ", replydate=CURDATE() where patientId="jj999 " and aid=41
If the user choose to accept the offer the message will be shown:

Successfully update the status!

Go back to your appointments, click here

Update information
The user will be able to update their information (partial information will be updated if only part of the field is filled)
(1) select name, ssn, birthday, address, latitude, longitude, phone, email, password from patient where patientId = "jj999 "
(2) the result variables will be replaced with filled content if the corresponding field is filled
(3) update patient set name=?, ssn=?, birthday=?, address=?, latitude=?, longitude=?, phone=?, email=?, password=? where patientId=?

Enter your information below:

Name:
SSN:
Birthday: mm / dd / yyyy
address:
phone: 12345678
email:
Password:
Submit

Go back

Update Preference
The patient can view/delete/add preferences for their vaccination appointment.

Weekday	TimeSlot	Distance	Delete
Sunday	08:00:00	10	Delete
Sunday	12:00:00	10	Delete
Wednesday	08:00:00	10	Delete
Wednesday	12:00:00	10	Delete
Thursday	08:00:00	1	Delete
Thursday	12:00:00	1	Delete
Friday	08:00:00	3	Delete
Friday	12:00:00	3	Delete
Saturday	08:00:00	1	Delete
Saturday	12:00:00	1	Delete

Add a preference: click here, Go back

SQL query:
(1) Check if the preference is already exist
select * from preference where patientId = ? and weekday=? and slot=?
(2) Do the corresponding insertion/deletion
delete from preference where patientId=? and weekday=? and slot=?
insert into preference(patientId,weekday,slot,distance) values (?,?,?,?)

Add a new preference:

Choose weekday Sunday Choose time 8:00AM-12:00PM Distance in miles:
Submit

Back to view preference

Provider Guide

For logged in provider, it will show options for viewing a summary of the appointment, all of the appointment and information of assigned appointment and filter options. The provider can choose to add an appointment.
Example queries for summary for the provider:
(1) The count for each status of appointments
with t1 as (select status from appointment, offer where appointment.aid=offer.aid and providerId="walgreen1"),
t2 as (select count(*) as o_ct from t1 where status="offer"),
t3 as (select count(*) as a_ct from t1 where status="accepted"),
t4 as (select count(*) as d_ct from t1 where status="declined"),
t5 as (select count(*) as c_ct from t1 where status="cancelled"),
t6 as (select count(*) as m_ct from t1 where status="missed"),
t7 as (select count(*) as com_ct from t1 where status="complete")
select o_ct, a_ct, d_ct, c_ct, m_ct, com_ct from t2,t3,t4,t5,t6,t7

(2) All appointment
select adate, atime from appointment where providerId="walgreen1" order by adate, atime
(3) Summary of the assigned appointment
select offer.patientId, offer.aid, adate, atime, patient.name, patient.phone, offerdate, deadline, replydate, status from offer, appointment, patient where offer.aid=appointment.aid and offer.patientId=patient.patientId and providerId="walgreen1"

Summary

Offered Count	Accept Count	Declined Count	Cancelled Count	Missed Count	Complete Count
2	0	0	3	4	1

Your appointments

date	time
2021-03-22	09:00:00
2021-03-22	14:00:00
2021-03-25	09:00:00
2021-03-25	14:00:00
2021-03-30	09:00:00
2021-03-30	14:00:00
2021-04-01	09:00:00
2021-04-01	14:00:00
2021-04-05	09:00:00
2021-04-05	14:00:00
2021-04-09	09:00:00
2021-04-09	14:00:00
2021-04-13	09:00:00
2021-04-17	14:00:00
2021-04-21	09:00:00
2021-04-25	14:00:00
2021-05-03	09:00:00
2021-05-07	14:00:00
2021-05-11	09:00:00
2021-05-15	14:00:00
2021-05-19	09:00:00
2021-05-23	14:00:00
2021-05-27	09:00:00
2021-05-31	14:00:00
2021-06-04	09:00:00
2021-06-08	14:00:00
2021-06-12	09:00:00
2021-06-16	14:00:00
2021-06-18	14:00:00
2021-06-18	14:00:00

Add New

Your appointments' assignments

date	time	name	phone	assigned time	reply by	reply date	status	update
2021-03-25	09:00:00	Jones	2398928398	2021-03-10	2021-03-17	2021-03-15	missed	change
2021-03-30	09:00:00	Jones	2398928398	2021-03-15	2021-03-22	2021-03-21	missed	change
2021-04-01	09:00:00	Jones	2398928398	2021-03-20	2021-03-27	2021-03-25	cancelled	change
2021-04-13	09:00:00	Jones	2398928398	2021-03-20	2021-03-27	2021-03-25	cancelled	change
2021-04-21	09:00:00	Bob	4982909274	2021-04-10	2021-04-17	2021-04-16	missed	change
2021-04-21	09:00:00	Jones	2398928398	2021-03-20	2021-03-27	2021-03-25	cancelled	change
2021-05-11	09:00:00	Jones	2398928398	2021-03-20	2021-03-27	2021-03-25	complete	change
2021-05-19	09:00:00	Bob	4982909274	2021-05-05	2021-05-12	2021-05-10	missed	change
2021-05-27	09:00:00	Ron	3294890289	2021-05-18	2021-05-25		offer	change
2021-06-08	14:00:00	Gabriel	5829902749	2021-05-18	2021-05-25		offer	change

Filter the appointment:

Choose status offer Sort by Acceptance(Reply Date) Submit

Logout

Add New Appointment

Add a new appointment:

Date: mm/dd/yyyy
Time -- : -- Submit

Go back to view all appointment

The web application will first check if the date entered is at least one week from today and then perform the insertion:

insert into appointment(adate, atime, providerId) values (?,?,?)

One provider can have multiple appointments at the same time, the appointment will be assigned an aid when inserting into table “appointment”.

Filtering the “missed” appointment and sort them by appointment time:

select adate, atime, patient.name, patient.phone, offerdate, deadline, replydate, status from offer, appointment, patient where offer.aid=appointment.aid and offer.patientId=patient.patientId and providerId="walgreen1" and status="missed" order by adate, atime

date	time	name	phone	assigned time	reply by	reply date	status
2021-03-25	09:00:00	Jones	2398928398	2021-03-10	2021-03-17	2021-03-15	missed
2021-03-30	09:00:00	Jones	2398928398	2021-03-15	2021-03-22	2021-03-21	missed
2021-04-21	09:00:00	Bob	4982909274	2021-04-10	2021-04-17	2021-04-16	missed
2021-05-19	09:00:00	Bob	4982909274	2021-05-05	2021-05-12	2021-05-10	missed

Go back

Administrator Interface

Click here to view patients and edit their group.

Click here to assign appointment to users.

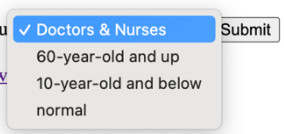
The administrator will be able to check each patient’s information and change their priority

select patientId, name, ssn, birthday, address, phone, email, patient.prionum, startdate from patient left join grouptype on patient.prionum=grouptype.prionum

Patient ID	Name	SSN	Birthday	Address	Phone	Email	Group number	Eligible Date	Change Group
bob0823	Bob	293019374	1983-08-23	824 Park Ave, Brooklyn, NY	4982909274	huanyao0718@gmail.com	1	2021-03-15	change
gabriel1105	Gabriel	629204862	1970-11-05	390 Gates Ave, Brooklyn	5829902749	huanyao0718@gmail.com	4	2021-06-01	change
jj999	Jack	628374018	1969-10-09	105 E 22nd St,New York, NY	8207468290	huanyao0718@gmail.com	4	2021-06-01	change
jones90	Jones	829380921	1990-09-02	343 Gold Street, Brooklyn, NY	2398928398	huanyao0718@gmail.com	1	2021-03-15	change
mary2021	Mary	108374928	2021-02-03	22 E 119th St, New York, NY	2019284038	huanyao0718@gmail.com	3	2021-05-15	change
michael6	Michael	820471928	1950-12-11	138-18 Northern Blvd, Flushing, NY	2890928392	huanyao0718@gmail.com	2	2021-04-30	change
ronron0607	Ron	402948465	1959-06-07	163 Eastern Pkwy, Brooklyn, NY	3294890289	huanyao0718@gmail.com	2	2021-04-30	change
rose1999	Rose	820174927	1999-03-02	1035 Washington Ave, Brooklyn, NY	8019309210	huanyao0718@gmail.com	4	2021-06-01	change
test0	Huanyao Ye	12121212	2021-05-07	343 Gold Street	6085040737	huanyao0718@gmail.com	1	2021-03-15	change
test1	HELLO	12344567	2021-01-15	343 Gold Street	6085040737	huanyao0718@gmail.com			change

Go back

Change group:

Choose Grou 

Submit

Back to view

Except from running the matching algorithm every hour, the administrator will be able to manually run algorithm to match the patients and appointments and send confirmation to the patients' email. The details of the algorithm is stated in the next section.

You just update the appointment offers!
Hello test1! You are assigned to the Flushing Hospital Medical Center at 4500 Parsons Blvd, Flushing, NY on 2021-06-14 08:00:00.

Patient ID	Offered Date	Reply By	Reply date	Status	Appointment ID	Appointment Date	Appointment Date	Provider ID
jones90	2021-03-10	2021-03-17	2021-03-15	missed	3	2021-03-25	09:00:00	walgreen1
jones90	2021-03-15	2021-03-22	2021-03-21	missed	5	2021-03-30	09:00:00	walgreen1
jones90	2021-03-20	2021-03-27	2021-03-25	cancelled	7	2021-04-01	09:00:00	walgreen1
jones90	2021-03-20	2021-03-27	2021-03-25	cancelled	13	2021-04-13	09:00:00	walgreen1
jones90	2021-03-20	2021-03-27	2021-03-25	cancelled	15	2021-04-21	09:00:00	walgreen1
jones90	2021-03-20	2021-03-27	2021-03-25	complete	19	2021-05-11	09:00:00	walgreen1
bob0823	2021-04-10	2021-04-17	2021-04-16	missed	15	2021-04-21	09:00:00	walgreen1
bob0823	2021-04-20	2021-04-27	2021-04-24	missed	36	2021-05-05	11:00:00	nycwoodhull
bob0823	2021-05-05	2021-05-12	2021-05-10	missed	21	2021-05-19	09:00:00	walgreen1
mary2021	2021-05-12	2021-05-19	2021-05-15	accepted	67	2021-05-22	14:00:00	cvs1
jj999	2021-05-15	2021-05-22	2021-05-16	cancelled	72	2021-07-22	10:00:00	cvs1
jj999	2021-05-15	2021-05-22	2021-05-16	cancelled	73	2021-07-22	14:00:00	cvs1
jj999	2021-05-15	2021-05-22	2021-05-16	cancelled	74	2021-07-24	10:00:00	cvs1
jj999	2021-05-15	2021-05-22	2021-05-18	cancelled	75	2021-07-24	14:00:00	cvs1
ronron0607	2021-05-18	2021-05-25		offer	23	2021-05-27	09:00:00	walgreen1
gabriel1105	2021-05-18	2021-05-25		offer	26	2021-06-08	14:00:00	walgreen1
jj999	2021-05-18	2021-05-25	2021-05-18	accepted	41	2021-06-10	11:00:00	nycwoodhull
michael6	2021-05-18	2021-05-25		offer	59	2021-05-29	15:00:00	walgreen2
rose1999	2021-05-18	2021-05-25		offer	75	2021-07-24	14:00:00	cvs1
test1	2021-05-18	2021-05-25		offer	115	2021-06-14	08:00:00	flushinghospital

Assign Appointment:
Update
Go back

The corresponding user will receive the email

(c) Matching algorithm

```
with t1 as(
        select
aid,adate,atime,providerId,test.patient.prionum,email,slot,test.patient.patientId,dist
ance
        from test.preference, test.appointment, test.patient, test.grouptype
        where weekday(adate)=test.preference.weekday and
        test.patient.patientId=test.preference.patientId and
        test.patient.prionum=test.grouptype.prionum and
        test.grouptype.startdate <= adate and
        adate > DATE_ADD(CURDATE(), INTERVAL 7 DAY) and
        TIMESTAMPDIFF(hour, slot, atime) >= 0 and
        TIMESTAMPDIFF(hour, slot, atime)<4),
      t2_1 as (
        select * from t1
        where aid not in (select aid from test.offer where status='offer' or
status='accepted' or status='complete') and
        patientId not in (select patientId from test.offer where status='offer' or
status='accepted' or status='complete')
      ),
      t2_2 as (
        select t2_1.aid, t2_1.patientId from t2_1, test.offer where t2_1.aid=offer.aid
and t2_1.patientId=offer.patientId and
        (status='missed' or status='declined' or status='cancelled')
      ),
      t2 as (
        select
t2_1.aid,t2_1.adate,t2_1.atime,t2_1.providerId,t2_1.prionum,t2_1.email,t2_1.slot,t2_1.
patientId,t2_1.distance
        from t2_1 left join t2_2 on t2_1.aid=t2_2.aid and
t2_1.patientId=t2_2.patientId where
        t2_2.aid is null or t2_2.patientId is null
      ),
      t3 as (
        select aid, adate, atime, t2.providerId, provider.name, provider.address,
t2.patientId, t2.prionum, t2.email,distance,
        111.111 * 0.62
        * DEGREES(ACOS(LEAST(1.0, COS(RADIANS(test.provider.latitude))
        * COS(RADIANS(test.patient.latitude))
        * COS(RADIANS(test.provider.longitude - test.patient.longitude))
        + SIN(RADIANS(test.provider.latitude))
        * SIN(RADIANS(test.patient.latitude))))) AS distance_in_miles
        from t2, test.provider, test.patient
        where test.provider.providerId=t2.providerId and
        test.patient.patientId=t2.patientId),
      t4 as (
        select patientId, prionum, email, providerId, name, address, aid, adate,
atime, distance_in_miles from t3
        where distance_in_miles <= distance
      ),
      t5 as (
        select aid, count(*) as a_ct from t4 group by aid
      )
      select patientId, prionum, email, providerId, name, address, t4.aid, adate,
atime, distance_in_miles
      from t4, t5 where t4.aid=t5.aid
      order by prionum, a_ct, adate, atime, distance_in_miles
```

(1) The first table joins patients' information with their group number to get the eligible date and join with preferences on patients' ID number. It also joins with appointment and the results which have the appointments matches in weekday, time slot and at least a week from today are chosen.
(2) The second table filtered out the appointment that has been offered/accepted/completed. In addition, if the patient already has an offered/accepted/completed appointment, he/she will also be filtered out and not participate in the following steps.
(3) If the patient once cancelled/missed/declined an appointment, then this appointment will not be assigned to the same patient.
(4) Calculate the distance from patient's address to provider's address, if the distance is greater than the corresponding preferred distance, it will also be filtered out.
(5) For each of the appointment in the table we have for now, compute how many patient is competing for this appointment.
(6) Sort the result by group priority number, the count of patients who are competing for this appointment, appointment date, appointment time and distance.
(7) Insert the information in first row of the table
(8) Repeat step 1-7 until there is no record in the table

Explanation of the algorithm:

For step 1~4, they are filtering the appointments that matches patients' schedule, preferred distance and eligible date. The appointment date must be greater than the eligible date, at least one week from the assigned date (today), and is not currently assigned to any patient. In addition, if the patient currently is assigned (offered/accepted/complete) an appointment, he/she will not participate in this process, and the cancelled/declined/missed appointment will not be assigned to the same patient.

The result table is sorted by group priority number, the count of patients who are competing for this appointment, appointment date, appointment time and distance. The table is sorted in this sequence is that the patient with higher priority number will be considered to assigned the appointment. Then the table will sort the appointment by the count of patients who are competing for this appointment, for example, if a patient is eligible for appointment A and appointment B, and 1 person is competing for the appointment A while 2 person is competing for the appointment. If the patient is assigned appoint A, then there will be only one person who are competing for the appointment B. Therefore, sorting by the count of patients who are competing for this appointment can assigned the appointments which are competed by less patients first so the count of patients who compete for other appointment decreases. Finally the result is sort by appointment date, appointment time and distance.

After The first record in the table is fetched, the corresponding patient will be offered corresponding appointment, and he/she will not participate when we run the query again. Then we repeat the step above and fetch the first record each time until there is no record in the table. In other words, all patients who originally has not been assigned an appointment and is eligible for some appointment now is offered one appointment.