(a) Users(uid, uname, email, password, state)

Primary key: uid

Note(nid, uid, text, tags, lid, radius, sid, view)

State is the user's current state like "at work" or "happy".

Primary key: nid

Foreign key: Users(uid)

Location(lid)

Schedual(sid)

View represents current note's visible range like "all" and "friends" and "me". It means current note is visible to all users in system or the user's friends who posts the note or just to itself.

Filter (fid, uid, lid, state, tag, sid)

Primary key: fid

Foreign key: Users(uid)

Location(lid)

Schedual(sid)

Friends(uid1, uid2)

Primary key: uid1,uid2

Foreign key: Users(uid1)

Users(uid2)

Comments(nid, uid, time, comment)

Primary key:nid, uid, time

Foreign key:Note(nid)

Users(uid)

Location(lid, latitude, longitude, name)

Primary key: lid

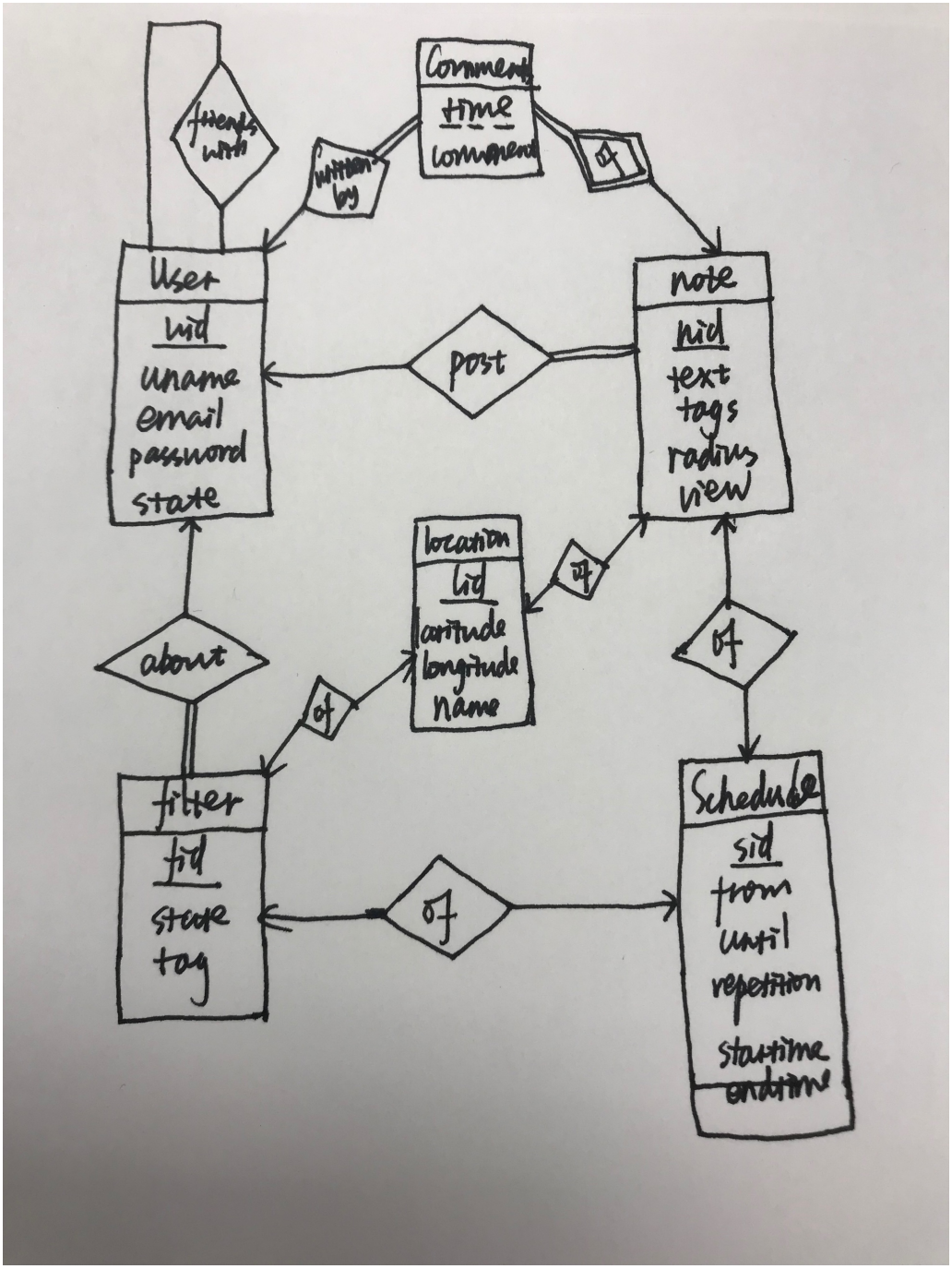
In this table, "latitude" and "longitude" shows a place's exact location, and name represent place name like "SoHo".

Schedual(sid, from, until, repetition, starttime, endtime)

Primary key: sid

In this table, datatype of “from” is datetime, the same with “until”, these two attribute show the period of this schedual, “repetition” represents repetition time like “ every Friday”, “starttime” and “endtime” show the time period in one day.

ER diagram：



(b)The structure of data shows below:

```
create table users(
    uid int,
    uname varchar(255),
    email varchar(255),
    password varchar(255),
    state varchar(255),
    primary key(uid)
);
create table location(
    lid int,
    latitude decimal(12,8),
    longitude decimal(12,8),
    primary key(lid)
);
create table schedual(
    sid int,
    from1 datetime,
    until datetime,
    repetition varchar(255),
    starttime time,
    endtime time
);
create table note(
    nid int,
    uid    int,
    text varchar(255),
    tags varchar(255),
    lid int,
    radius int,
```

```
    sid int,
    view varchar(255),
    primary key(nid),
    foreign key(uid) references users(uid),
    foreign key(lid) references location(lid),
    foreign key(sid) references schedual(sid)
);
create table filter(
    fid int,
    uid int,
    lid int,
    state varchar(255),
    tag varchar(255),
    sid    int,
    primary key(fid),
    foreign key(uid) references users(uid),
    foreign key(lid) references location(lid),
    foreign key(sid) references schedual(sid)
);
create table friends(
    uid1 int,
    uid2 int,
    primary key(uid1,uid2),
    foreign key(uid1) references users(uid),
    foreign key(uid2) references users(uid)
);
create table comments(
    nid int,
    uid int,
    time    datetime,
```

comment varchar(255),
primary key(nid,uid,time),
foreign key(nid) references note(nid),
foreign key(uid) references users(uid)
);

(c)

(1)insert into users(uid,uname,email,password,state) values (7,"Sum","007@gmail.com",
"77777","at work");

```
mysql> insert into users(uid,uname,email,password,state) values
    -> (7,"Sum","007@gmail.com","77777","at work");
Query OK, 1 row affected (0.08 sec)

mysql> select * from users;
+-----+--------+---------------+----------+---------------+
| uid | uname  | email         | password | state         |
+-----+--------+---------------+----------+---------------+
|   1 | Jack   | 001@gmail.com | 11111    | at work       |
|   2 | Linda  | 002@gmail.com | 22222    | lunch break   |
|   3 | Hanna  | 003@gmail.com | 33333    | at home       |
|   4 | Lucy   | 004@gmail.com | 44444    | just chilling |
|   5 | Steven | 005@gmail.com | 55555    | just chilling |
|   6 | Amy    | 006@gmail.com | 66666    | at work       |
|   7 | Sum    | 007@gmail.com | 77777    | at work       |
+-----+--------+---------------+----------+---------------+
7 rows in set (0.00 sec)
```

(2) Insert into note(nid,uid,text,tags,lid,radius,sid) values (7,6,"text6","shopping",5,120,4);

```
mysql> Insert into note(nid,uid,text,tags,lid,radius,sid) values
    -> ("000007","06","text6","shopping","005","120","0004");
Query OK, 1 row affected (0.10 sec)

mysql> select * from note;
+-----+------+-------+---------------+------+--------+------+
| nid | uid  | text  | tags          | lid  | radius | sid  |
+-----+------+-------+---------------+------+--------+------+
|   1 |    1 | text1 | food          |    1 |     50 |    2 |
|   2 |    3 | text2 | food          |    4 |     75 |    1 |
|   3 |    5 | text3 | shopping      |    2 |     95 |    3 |
|   4 |    2 | text4 | tourism       |    5 |     30 |    5 |
|   5 |    4 | text5 | transportation |    3 |     60 |    4 |
|   6 |    1 | text6 | me            |    2 |    100 |    2 |
|   7 |    6 | text6 | shopping      |    5 |    120 |    4 |
+-----+------+-------+---------------+------+--------+------+
7 rows in set (0.00 sec)
```

(3) select uid1,uid2 from friends where uid1 = '3' or uid2 = '3';

```
mysql> select uid1,uid2 from friends
    -> where uid1 = '3' or uid2 = '3';
+------+------+
| uid1 | uid2 |
+------+------+
|    1 |    3 |
|    3 |    4 |
|    3 |    6 |
+------+------+
3 rows in set (0.00 sec)
```

(4) select n.nid

from note n, filter f, location l1, lcation l2, note n, schedua s,friends where f.uid="000001" and f.state="at work" and f.sid=s.sid and l1.lid=f.lid and

s.from<="2018-10-22" and s.until>="2018-10-22" and s.starttime<="12:00" and s.endtime>="12:00"

and n.view="all" or(n.view="me" and n.uid="000001")or(n.view="friend" and n.uid in friends.uid1 and friends.uid2="000001")

and n.lid =l2.lid and dist(l1,l2)<n.radius and n.tag=f.tag;

In this query, I assume current time is “2018-10-22 12:00” and giveng user’s id is 1 and his state is “at work”, so firstly I look up in the filter

table to find which kinds of notes he wants to receive, then I look up in the note table to find those notes which is visible to the user now and whose tag is the tag he wants to receive and I also need to check the distance of user's current location and note's location, I use "dist" to calculate that. There are three kinds of notes visible to given user, that is visible to all users in the system or is given user's friends' note which is posted to friends or is those note given user post to himself.

(5) select f.uid

from note n, schedual s, filter f, user u, location l1, location l2

where n.nid=1and f.sid=s.sid and s.from<="2018-10-22" and s.until>="2018-10-22" and s.starttime<="12:00" and s.endtime>="12:00"

and u.uid=f.uid and u.state=f.state and n.lid=l1.lid and f.lid=l2.lid and dist(l1,l2)<radius and f.tag=n.tag;

In this query, I also assume current time is "2018-10-22 12:00" and given note's id =1, I need to check who can receive this note now in filter table. The query process is similar with the last question.

(6) select n.nid

from note n, filter f, location l1, lcation l2, note n, schedua s,friends

where f.uid="000001" and f.state="at work" and f.sid=s.sid and l1.lid=f.lid and

s.from<="2018-10-22" and s.until>="2018-10-22" and s.starttime<="12:00" and s.endtime>="12:00"

and n.view="all" or(n.view="me" and n.uid="000001")or(n.view="friend" and n.uid in friends.uid1 and friends.uid2="000001")

and n.lid =l2.lid and dist(l1,l2)<n.radius and n.tag=f.tag and text like "good%";

This query is similar with the fourth query, but it is possible that there are a lot of results in forth query, so I can add a condition that the " text"

must start with “good” using “like” operator.

(d)We have inserted some data into every table, the results show below:

```
mysql> select * from users;
+-----+--------+---------------+----------+---------------+
| uid | uname  | email         | password | state         |
+-----+--------+---------------+----------+---------------+
|   1 | Jack   | 001@gmail.com | 11111    | at work       |
|   2 | Linda  | 002@gmail.com | 22222    | lunch break   |
|   3 | Hanna  | 003@gmail.com | 33333    | at home       |
|   4 | Lucy   | 004@gmail.com | 44444    | just chilling |
|   5 | Steven | 005@gmail.com | 55555    | just chilling |
|   6 | Amy    | 006@gmail.com | 66666    | at work       |
+-----+--------+---------------+----------+---------------+
```

```
mysql> select * from location;
+-----+------------+-------------+-------------+
| lid | lname      | latitude    | longitude   |
+-----+------------+-------------+-------------+
|   1 | soho       | 120.00000000 | 130.00000000 |
|   2 | Disney     | 160.00000000 | 140.00000000 |
|   3 | Flushing   |  50.00000000 |  90.00000000 |
|   4 | Newport    |  75.00000000 |  60.00000000 |
|   5 | LongIsland | 110.00000000 |  70.00000000 |
+-----+------------+-------------+-------------+
```

```
mysql> select * from note;
+-----+------+---------------+---------------+------+--------+------+---------+
| nid | uid  | text          | tags          | lid  | radius | sid  | view    |
+-----+------+---------------+---------------+------+--------+------+---------+
|   1 |    1 | text1         | food          |    1 |     50 |    2 | all     |
|   2 |    3 | text2         | food          |    4 |     75 |    1 | all     |
|   3 |    5 | text3         | shopping      |    2 |     95 |    3 | friends |
|   4 |    2 | text4         | tourism       |    5 |     30 |    5 | friends |
|   5 |    4 | text5         | transportation |    3 |     60 |    4 | me      |
|   6 |    1 | text6         | me            |    2 |    100 |    2 | all     |
|   7 |    1 | good for lunch | food          |    5 |    200 |    5 | friend  |
+-----+------+---------------+---------------+------+--------+------+---------+
```

```
mysql> select * from filter;
+-----+------+------+--------------+----------------+------+
| fid | uid  | lid  | state        | tag            | sid  |
+-----+------+------+--------------+----------------+------+
|   1 |    1 |    1 | at work      | food           |    2 |
|   2 |    3 |    2 | at home      | shopping       |    3 |
|   3 |    2 |    5 | lunch break  | tourism        |    5 |
|   4 |    5 |    4 | just chilling | transportation |    4 |
|   5 |    4 |    5 | just chilling | food           |    5 |
+-----+------+------+--------------+----------------+------+
```

```
mysql> select * from friends;
+------+------+
| uid1 | uid2 |
+------+------+
|    1 |    2 |
|    1 |    3 |
|    3 |    4 |
|    2 |    5 |
|    4 |    5 |
|    1 |    6 |
|    3 |    6 |
+------+------+
```

```
mysql> select * from comments;
+-----+-----+---------------------+----------+
| nid | uid | time                | comment  |
+-----+-----+---------------------+----------+
|   1 |   2 | 2018-05-01 00:00:00 | comment1 |
|   2 |   3 | 2017-12-26 00:00:00 | comment2 |
|   3 |   1 | 2018-03-02 00:00:00 | comment3 |
|   4 |   5 | 2018-07-31 00:00:00 | comment4 |
|   5 |   6 | 2018-09-06 00:00:00 | comment5 |
+-----+-----+---------------------+----------+
```