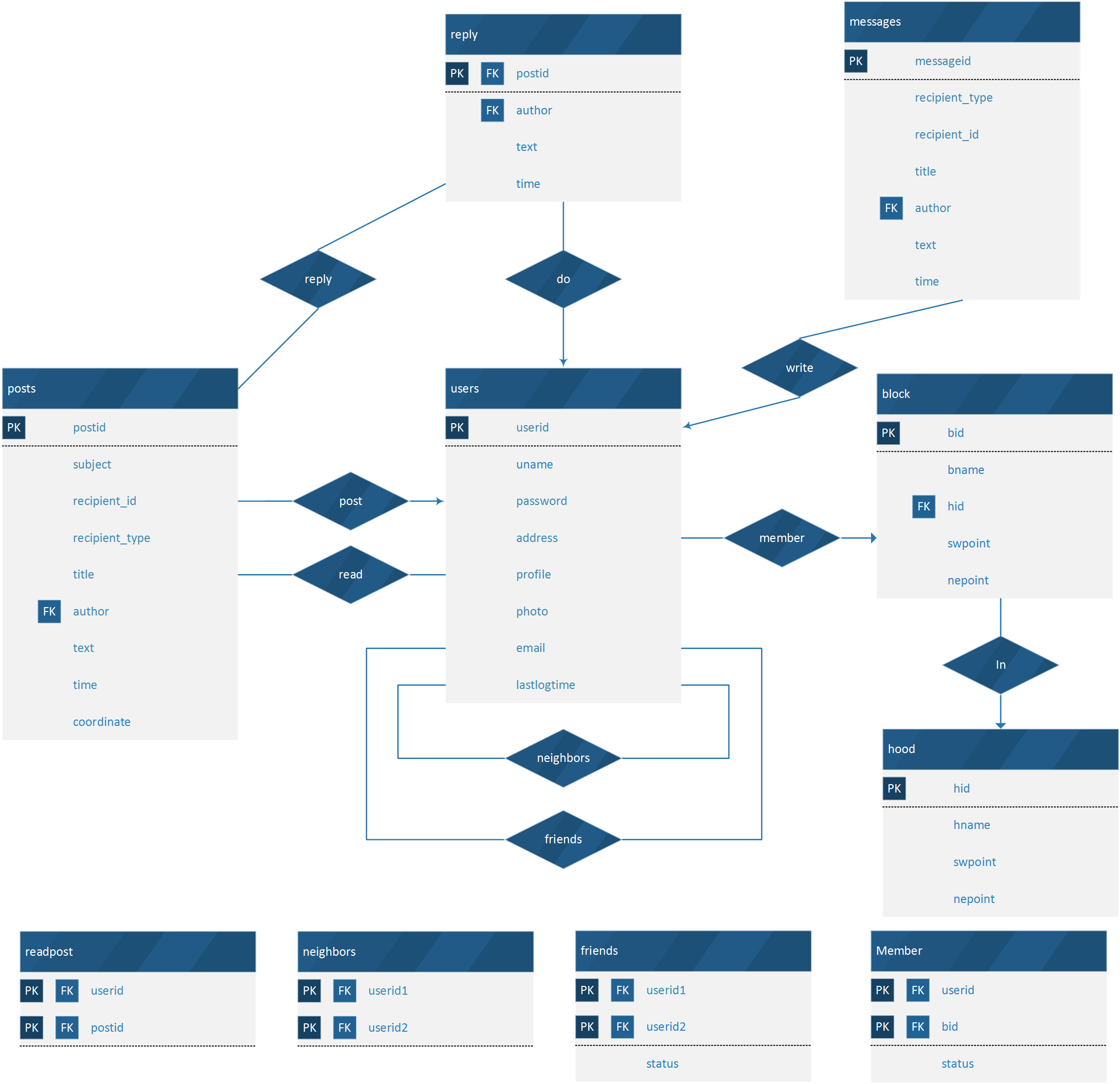
**Project #1**

**Hao Yu hy987, Wei Hong wh951**

(a)

**ER diagram**



**relational schema**

users (`userid`, `uname`, `password`, `address`, `profile`, `photo`, `email`, `lastlogtime`)

PRIMARY KEY (`userid`)

`hood` (`hid`,`hname`,`swpoint`,`nepoint`)

PRIMARY KEY (`hid`)

`block` (`bid`,`hid`,`bname`,`swpoint`,`nepoint`)

PRIMARY KEY (`bid`)

`member` (`userid`,`bid`,`status`)

PRIMARY KEY (`userid`)

FOREIGN KEY (`userid`) REFERENCES `users` (`userid`)

FOREIGN KEY (`bid`) REFERENCES `block` (`bid`)

`friends` (`userid1`,`userid2`,`status`)

PRIMARY KEY (`userid1`,`userid2`)

FOREIGN KEY (`userid1`) REFERENCES `users` (`userid`)

FOREIGN KEY (`userid2`) REFERENCES `users` (`userid`);

`neighbors` (`userid1`,`userid2`)

PRIMARY KEY (`userid1`,`userid2`)

FOREIGN KEY (`userid1`) REFERENCES `users` (`userid`)

FOREIGN KEY (`userid2`) REFERENCES `users` (`userid`)

`messages` (`messageid`,`recipient\_type`,`recipient\_id`,`title`,`author`,`text`,`time`)

PRIMARY KEY (`messageid`)

FOREIGN KEY (`author`) REFERENCES `users` (`userid`)

`posts` (`postid`,`subject`,`recipient\_type`,`recipient\_id`,`title`,`author`,`text`,`time`,`coordinate`)

PRIMARY KEY (`postid`)

FOREIGN KEY (`author`) REFERENCES `users` (`userid`)

`readpost` (`userid`,`postid`)

PRIMARY KEY (`userid`,`postid`)

FOREIGN KEY (`userid`) REFERENCES `users` (`userid`)

FOREIGN KEY (`postid`) REFERENCES `posts` (`postid`)

`reply` (`postid`,`replier`,`text`,`time`)

PRIMARY KEY (`postid`,`replier`,`time`)

FOREIGN KEY (`replier`) REFERENCES `users` (`userid`)

FOREIGN KEY (`postid`) REFERENCES `posts` (`postid`)

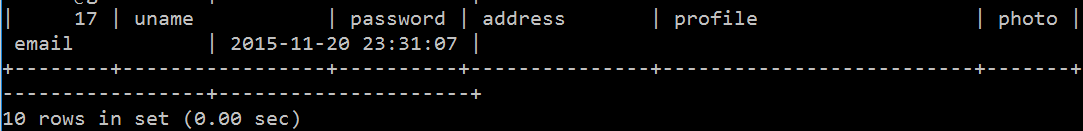
(c)

**queries and results**

1. Joining

**signs up:**

insert into users (uname, password, address, profile, photo, email) values ('uname', 'password', 'address', 'profile', 'photo', 'email');

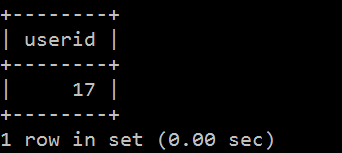


**apply to a block, first select the userid corresponding to user’s unique email then insert into member table and wait for the current member to approve:**

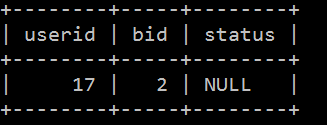
select userid

from users

where email = 'email';



insert into member values (17, 2, NULL);

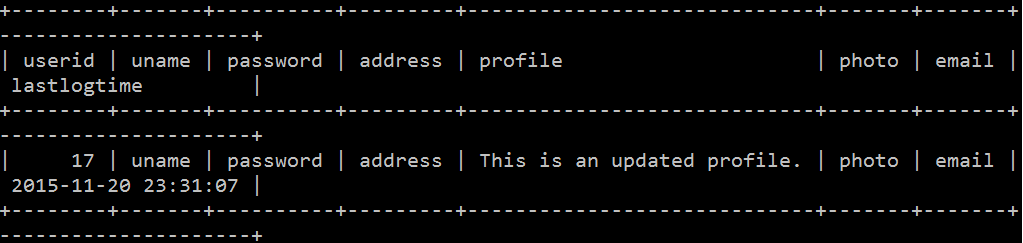


update profile:

update users

set profile = 'This is an updated profile.'

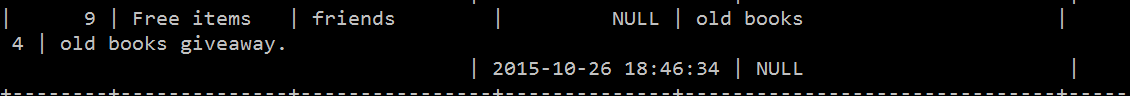
where userid = 17;



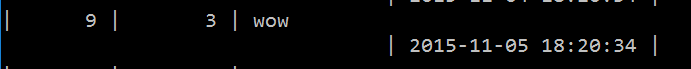
1. Content posting

contents are categorized into 3 types: post, replies to a post, (private) message.

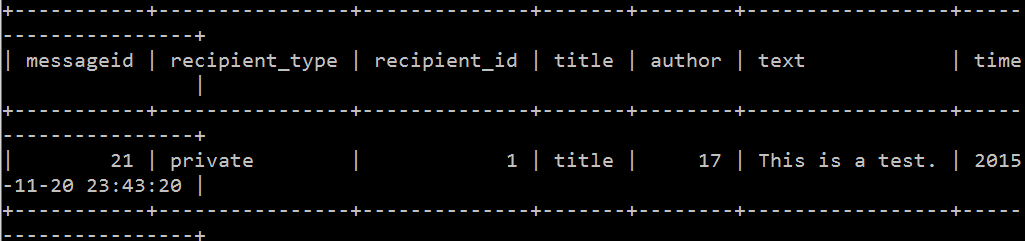
insert into posts (subject,recipient\_type,recipient\_id,title,author,text,time) values ('Free items','friends',NULL,'old books',4,'old books giveaway.', '2015-11-05 12:46:34');



insert into reply (postid,replier,text) values (9,3,'wow');



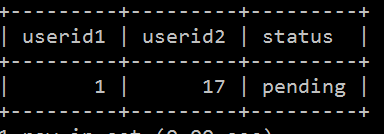
insert into messages (recipient\_type, recipient\_id, title, author, text) values ('private', '1', 'title', '17', 'This is a test.');



1. Friendship

**send and accept a friend request:**

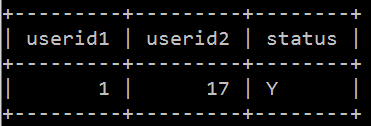
insert into friends (userid1,userid2) values ('1', '17');



update friends

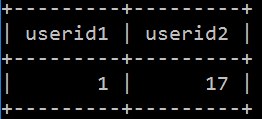
set status = 'Y'

where userid1 = '1' and userid2 = '17';



**add a neighbor:**

insert into neighbors (userid1,userid2) values ('1', '17');



**select a user’s friends and neighbors:**

select uname

from users

where userid in (

select userid2

from friends

where userid1 = '3')

or userid in (

select userid1

from friends

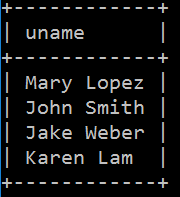
where userid2 = '3')

or userid in (

select userid2

from neighbors

where userid1= '3');



1. Browse and Search Messages

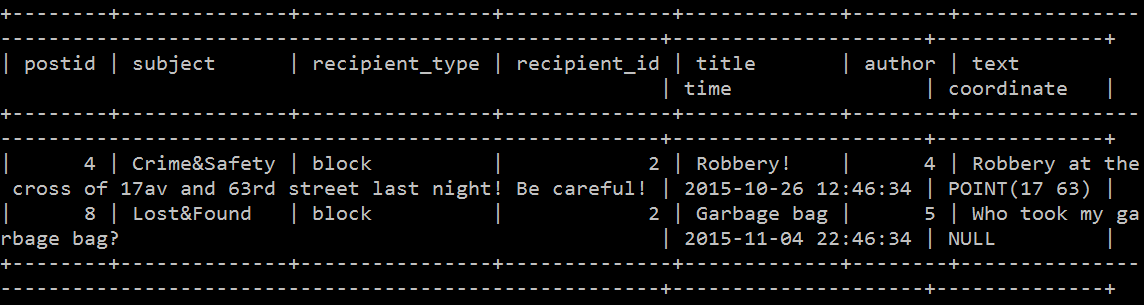
**select all the block feeds since the user’s last logout:**

select distinct(posts.postid),subject,recipient\_type,recipient\_id,title,author,posts.text,posts.time,AsText(coordinate) as coordinate

from posts join reply on posts.postid = reply.postid

where reply.time > (select lastlogtime from users where userid = 3)

and recipient\_type = 'block' and recipient\_id = (select bid from member where userid = 3);



**select all the unread friend feeds:**

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

where (author in (

select userid2

from friends

where userid1 = '2')

or author in (

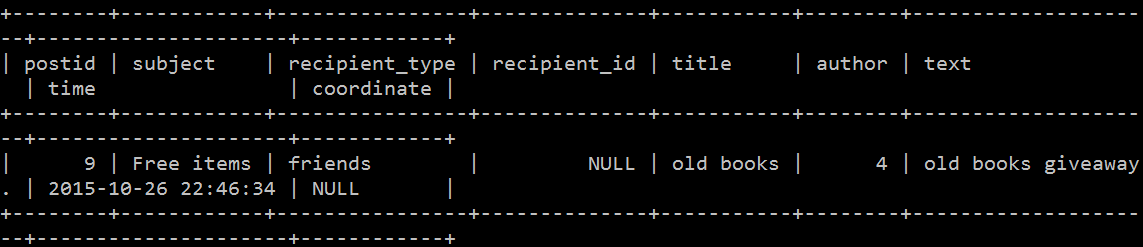
select userid1

from friends

where userid2 = '2'))

and recipient\_type = 'friends'

and postid not in (select postid from readpost where userid = '2');



**select all the posts accessible to the users that contains the keyword sale:**

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

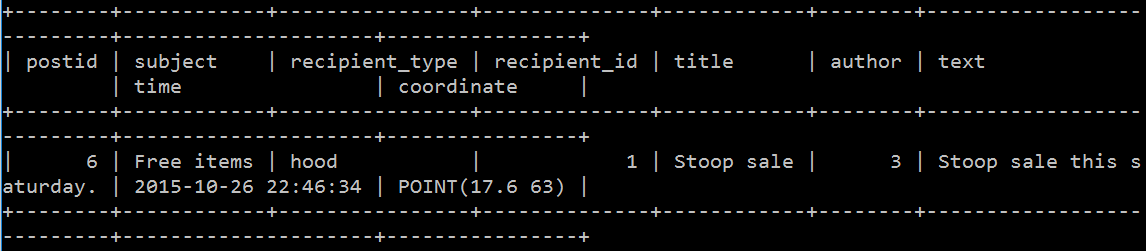
where text like "%sale%"

and (recipient\_type = 'block' and recipient\_id = (select bid from member where userid = 1)

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = 1))

or (recipient\_type = 'neighbors' and 1 in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (1 in (select userid1 from friends where userid2 = author) or 1 in (select userid2 from friends where userid1 = author))));



(d)

**test data illustration**

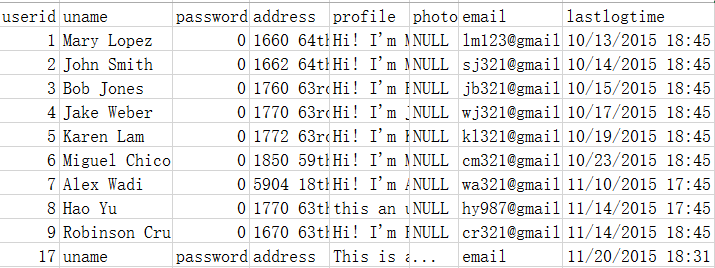
hood

|  |  |  |  |
| --- | --- | --- | --- |
| hid | hname | swpoint | nepoint |
| 1 | Bensonhurst West | point(18,62) | point(16,70) |
| 2 | Bensonhurst East | point(20,58) | point(18,62) |

blocks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bid | hid | bname | swpoint | nepoint |
| 1 | 1 | 16th Avenue 62-64 Street | point(17,62) | point(16,64) |
| 2 | 1 | 17th Avenue 62-64 Street | point(18,62) | point(17,64) |
| 3 | 2 | 18th Avenue 58-60 Street | point(20,58) | point(18,60) |

users



member

|  |  |  |
| --- | --- | --- |
| userid | bid | status |
| 1 | 1 | Y |
| 2 | 1 | Y |
| 3 | 2 | Y |
| 4 | 2 | Y |
| 5 | 2 | Y |
| 6 | 3 | Y |
| 7 | 3 | Y |
| 8 | 2 | 1 |
| 9 | 1 | NULL |
| 17 | 2 | NULL |

friends

|  |  |  |
| --- | --- | --- |
| userid1 | userid2 | status |
| 1 | 2 | Y |
| 1 | 3 | pending |
| 1 | 17 | Y |
| 2 | 3 | Y |
| 2 | 4 | Y |
| 3 | 4 | pending |
| 4 | 5 | Y |
| 6 | 7 | Y |

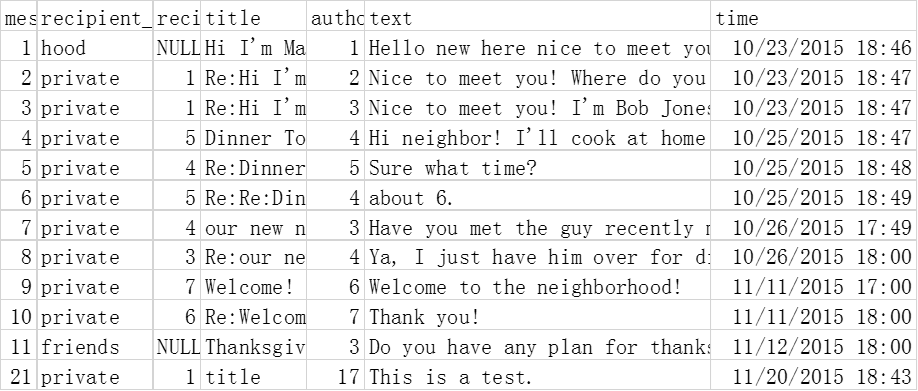
neighbors

|  |  |
| --- | --- |
| userid1 | userid2 |
| 2 | 1 |
| 1 | 2 |
| 4 | 3 |
| 3 | 4 |
| 3 | 5 |
| 6 | 7 |
| 1 | 17 |

posts



messages



reply



readpost



(e)

1. **Search the keywords through all the accessible posts and output the posts and the number of the posts containing the keywords. User can specify the field to search, such as subject, tile, text or all the above.**

use neighborhood;

drop procedure IF EXISTS postsSearcher;

DELIMITER //

CREATE PROCEDURE postsSearcher (IN uid int(10), searchType enum('subject','title','text','all'), keywords varchar(25), OUT totalNum int(10))

BEGIN

case searchType

when 'subject' then

BEGIN

set totalNum = (select count(\*)

from posts

where `subject` like CONCAT('%',keywords,'%') and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author)))));

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

where `subject` like CONCAT('%',keywords,'%')and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author))));

END;

when 'title' then

BEGIN

set totalNum = (select count(\*)

from posts

where `title` like CONCAT('%',keywords,'%') and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author)))));

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

where `title` like CONCAT('%',keywords,'%')and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author))));

END;

when 'text' then

BEGIN

set totalNum = (select count(\*)

from posts

where `text` like CONCAT('%',keywords,'%') and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author)))));

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

where `text` like CONCAT('%',keywords,'%')and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author))));

END;

when 'all' then

BEGIN

set totalNum = (select count(\*)

from posts

where (`subject` like CONCAT('%',keywords,'%') or `title` like CONCAT('%',keywords,'%') or `text` like CONCAT('%',keywords,'%')) and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author)))));

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

where (`subject` like CONCAT('%',keywords,'%') or `title` like CONCAT('%',keywords,'%') or `text` like CONCAT('%',keywords,'%'))and ((recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

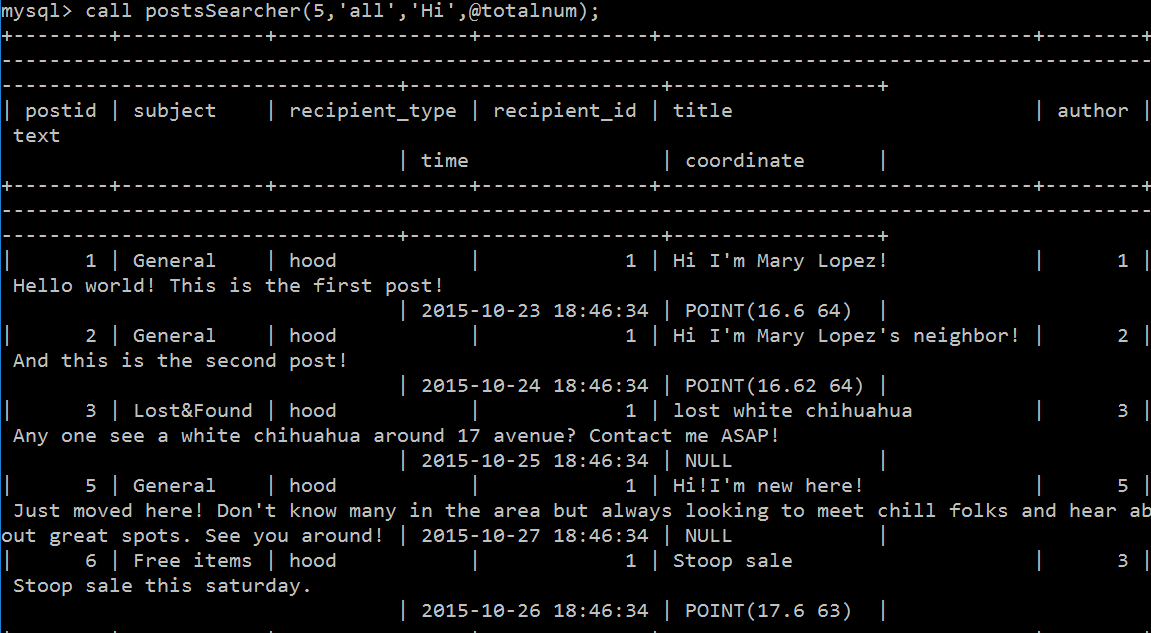
or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author))));

END;

END CASE;

END //

DELIMITER ;



2. **Retrieve all the accessible posts to the user.**

use neighborhood;

drop procedure IF EXISTS getReadablePost;

DELIMITER //

CREATE PROCEDURE getReadablePost (IN uid int(10))

BEGIN

select postid,subject,recipient\_type,recipient\_id,title,author,text,time,AsText(coordinate) as coordinate

from posts

where (recipient\_type = 'prviate' and recipient\_id = uid)

or (recipient\_type = 'block' and recipient\_id = (select bid from member where userid = uid))

or (recipient\_type = 'hood' and recipient\_id = (select block.hid from member natural join block join hood where block.hid = hood.hid and userid = uid))

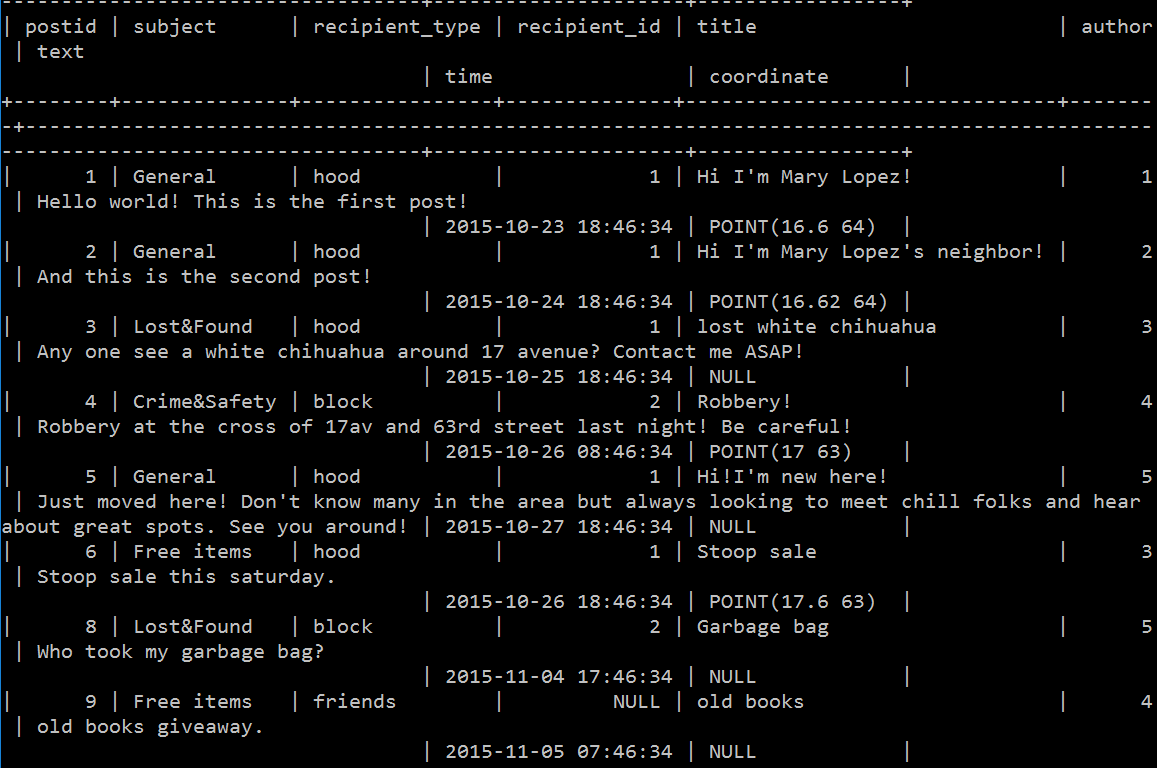
or (recipient\_type = 'neighbors' and uid in (select userid2 from neighbors where userid1 = author))

or (recipient\_type = 'friends' and (uid in (select userid1 from friends where userid2 = author) or uid in (select userid2 from friends where userid1 = author)));

END //

DELIMITER ;

call getReadablePost (5)



3. **Switch the userid1 with userid2 when inserting a tuple into friends and userid1 > userid2.**

DELIMITER //

create trigger friends\_check1 before insert on `friends`

for each row

begin

DECLARE userid INT;

set userid = new.userid2;

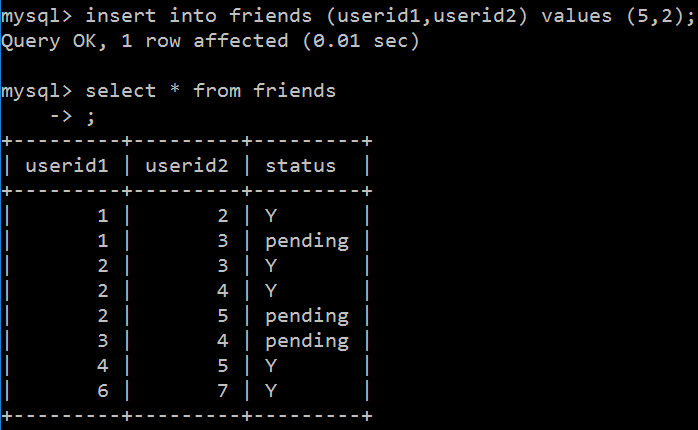
if new.userid1 > new.userid2 then

set new.userid2 = new.userid1, new.userid1 = userid;

end if;

end //

DELIMITER ;



4. **Membership update and insertion on a block with less than 3 members.**

use neighborhood;

drop trigger if exists member\_check;

drop trigger if exists member\_check2;

DELIMITER //

create trigger member\_check before update on `member`

for each row

begin

if (select count(\*)

from member

where bid = new.bid and status = 'Y' ) = new.status + 0 then

set new.status = 'Y';

end if;

end //

DELIMITER ;

DELIMITER //

create trigger member\_check2 before insert on `member`

for each row

begin

if (select count(\*)

from member

where bid = new.bid) < 1 then

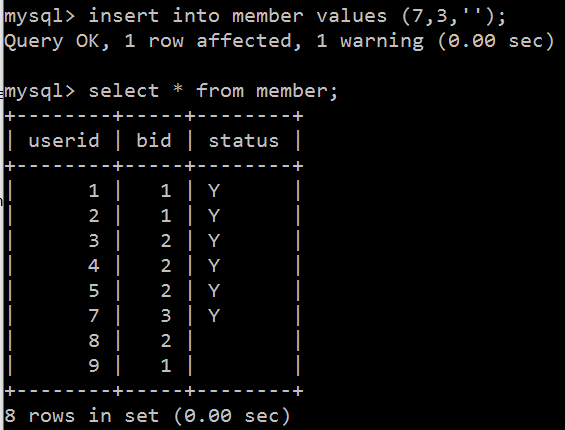
set new.status = 'Y';

end if;

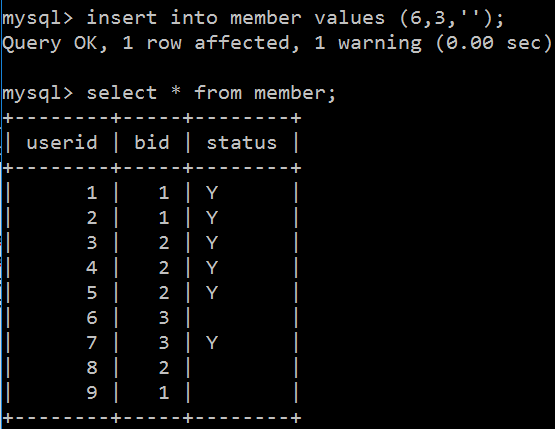
end //

DELIMITER ;

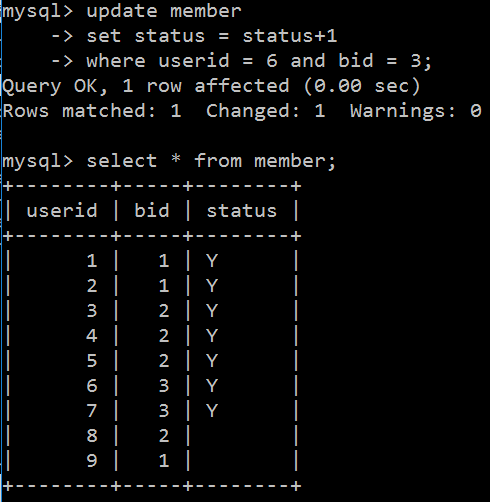
user 7 is the first member of block 3



user 6 then request to join block 3



user 7 then approves the request



(f)

**description**

**hood and block**

In our schema, the blocks and hoods are predefined and assigned to a unique id. Every block belongs to a hood. Blocks and hoods are rectangles defined by their southwest points and northeast points.

**users**

When a user registers in our system, the name, address, password, and a unique email are required. And a unique user id will be assigned to the user automatically. The user is not a member of any block at first. But after the user specifies his block, the system administrator will send messages to all the current members in the block. And a tuple with the user id and block id is inserted into member table but with status set to NULL. A member then choose yes or no to indicate whether to accept the user as a member. When there’re 3 members or all the members in a block when the number is less than 3 approve, the status of the corresponding tuple in the member table will be updated to ‘Y’, which suggests the membership is established. A user can update its profile in the system.

**relationship**

A user with membership can friend other users in the same hood and specify neighbors in the same block. To friend another user, the user select the person from a list which he can friend and then a system message will be sent to ask the person whether or not to accept the request. And a tuple of the pair of user id is inserted into the friends table and the status is defaulted to ‘pending’. Also a constriction that user id 1 must greater than user id 2 is imposed using trigger. Once the request is approved, the status is updated to ‘Y’ which indicates the friendship is valid. Otherwise, the tuple will be deleted. As for neighbors, the relationship is chosen unilaterally. A user can specify any user in the same hood as a neighbor. After a user moved to another hood, the neighbors of the user are lost while the user still keeps his friends.

**messages**

The messages in our system are categorized into 3 types. A post is posted to a group people (hood, block, friends, or neighbors). A reply is a reply to the post, which can be seen by all the people specified by the original post. A message is private message between two users while a sender can specify a group as recipient and unlike a post reply, a reply to a message is also a message and can only be seen by the sender and receiver. So the post is assigned a postid while reply only have the original postid and a time so as to be displayed in order of time. A post can be marked as read. The replies added after the time when the post is marked as read is regarded as unread reply. Similarly, a message has a messageid but not a subject.

**general use case**

A user use his email and password to log in. After that, the user is directed to a page presenting all the posts he can access. A user can start a new post after choose a subject from a predefined set of subjects and specify a title and type in text. Also user can review a post and reply. A user can do some operations on the posts, like search the keywords or categorize posts into feeds with different type of recipients. Messages are stored in a mailbox. After the user moves to another block, the messages are kept while the previous posts cannot be accessed any more. Every time a user logs out, the lastlogtime is updated in the users table.