

Question3:

Kim, Tae Young (tyk252)
Fang, Minda (mf3308)

In this part we use **MySQL** to solve the question.

For the two *friend.txt* and *like.txt* files, we delete the first line of the file and then import them into DB.

Create table ***friend***:

```
drop table if exists friend;

create table friend (
  person1 int,
  person2 int,
  index (person1),
  index (person2),
  index (person1, person2)
);
```

Create table ***liketable*** (notice that *like* is a reserved word in MySQL, so we choose the name *liketable*):

```
drop table if exists liketable;

create table liketable (
  person INT,
  artist INT,
  index (person),
  index (artist),
  index (person, artist)
);
```

Load table ***friend***:

```
LOAD DATA INFILE '/var/lib/mysql-files/friends.txt' INTO TABLE question3.friend
FIELDS
TERMINATED BY ','
```

Load table ***liketable***:

```
LOAD DATA INFILE '/var/lib/mysql-files/like.txt' INTO TABLE question3.liketable
FIELDS
TERMINATED BY ','
```

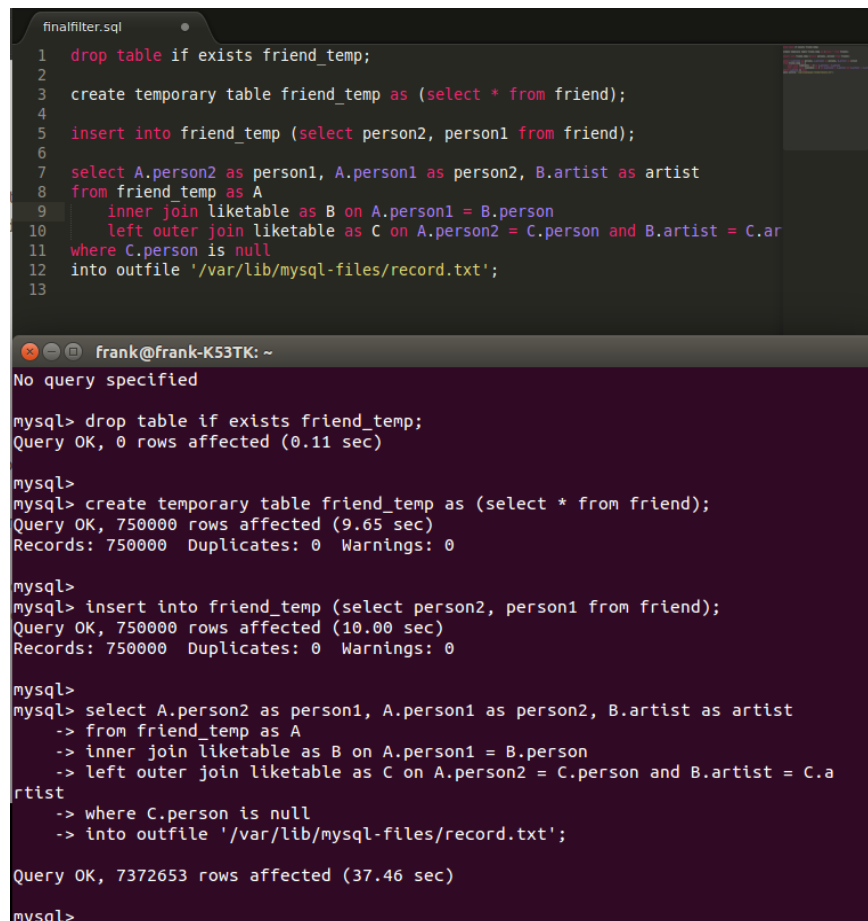
Function:

```
drop table if exists friend_temp;

create temporary table friend_temp as (select * from friend);

insert into friend_temp (select person2, person1 from friend);

select A.person2 as person1, A.person1 as person2, B.artist as artist
from friend_temp as A
    inner join liketable as B on A.person1 = B.person
    left outer join liketable as C on A.person2 = C.person and B.artist = C.artist
where C.person is null;
```



```
finalfilter.sql
1 drop table if exists friend_temp;
2
3 create temporary table friend_temp as (select * from friend);
4
5 insert into friend_temp (select person2, person1 from friend);
6
7 select A.person2 as person1, A.person1 as person2, B.artist as artist
8 from friend_temp as A
9     inner join liketable as B on A.person1 = B.person
10    left outer join liketable as C on A.person2 = C.person and B.artist = C.a
11 where C.person is null
12 into outfile '/var/lib/mysql-files/record.txt';
13

frank@frank-K53TK: ~
No query specified

mysql> drop table if exists friend_temp;
Query OK, 0 rows affected (0.11 sec)

mysql>
mysql> create temporary table friend_temp as (select * from friend);
Query OK, 750000 rows affected (9.65 sec)
Records: 750000 Duplicates: 0 Warnings: 0

mysql>
mysql> insert into friend_temp (select person2, person1 from friend);
Query OK, 750000 rows affected (10.00 sec)
Records: 750000 Duplicates: 0 Warnings: 0

mysql>
mysql> select A.person2 as person1, A.person1 as person2, B.artist as artist
-> from friend_temp as A
-> inner join liketable as B on A.person1 = B.person
-> left outer join liketable as C on A.person2 = C.person and B.artist = C.a
rtist
-> where C.person is null
-> into outfile '/var/lib/mysql-files/record.txt';

Query OK, 7372653 rows affected (37.46 sec)

mysql>
```

The idea first of all is to create a temporary table based on the “friend” data, which lists the identities of person1 and person2. Now, the relations among all persons are symmetric, that is the temporary table will not contain record duplicates. We can therefore use ordinary joins, first by applying an inner join between the temp table and the “like” data, which lists the identities of the artists and the persons, by filtering out the artists favored by a user’s friends . Then we apply an outer join, an outer left join specifically with the bigger subset as the criterion, to mark null in records where the user too lists that artist as its favorite. This query takes roughly 37 seconds, which isn’t too bad.

The output file(“**record.txt**”) is put in the same directory as this PDF file.