Introduction to Databases – Project1-part 3

Bin Qi (bq2130) Yunqiu Yao (yy2827)

The **URL** to access the application:
http:// 35.237.166.24:8111/

Link to the **GitHub repo** containing the codebase:
https://github.com/roqi/project1/tree/master/4111DB_project1

Changes made to the schema since Part 2: None

Application description:

General:
interactively display the Venmo user activities in our database, in addition to our risk analysis on Venmo transactions.

1) index.html / users.html / trxs.html / trx.html / user.html
   On index.html, the home page, we first set up two hyperlinks to present the whole user (users.html) and transaction (trxs.html) data stored in our database. Within trxs.html, by clicking on one specific date, one can view all the transactions happened on that day (trx.html); similarly, with in users.html, click on a user name and our app user can easily check that Venmo user's personal info (user.html)

   Also, we designed a multi-functional search window with three search keys: 'user name', 'risk level' and 'date'. To use this, first select one function within the second column, and then type in the corresponding elements within the first column's blank space.

2) User.html
   One search function can be realized by filing in a user name, e.g., Guo-Zhiqi. After clicking on the search query, a new page, user.html can be accessed. On this, not only is someone's personal info displayed, but also his bank account info and risk evaluation result. We conducted this risk analysis based on 4 aggregate results: number of public transactions, size of friends shown in transactions, comments, and likes, the maximum word length of transaction messages, and whether there is a track to someone's Facebook account. The risk level ranges from 0 to 4. A higher value indicates riskier Venmo account and more serious privacy leakage.

   Apart from that, a user could access a google search webpage based on the risk level result and the specific risks involved. We previously selected 16 subsets of keywords for all risk situations concerned, and the corresponding google search is based on these keywords.

Moreover, a hyperlink is placed to let an app user check risk evaluation results of other users(risk.html) who share the same risk level.

3) Risk.html
   This page explains in details risk dimensions and standards.
   One can directly access this page from home page, with the second search function, ‘search users by risk level (0-4).

4) Trx.html
   The purpose of this page is slight different from other pages mentioned, as we focus on transactions, not Venmo users.

   This page can be reached by the third search function on home page, i.e. to search for all the transactions happened on the same day. Then, same-day detailed transaction info will display.

In [1]:

```
ib.connect_db('ib://yy2827/W4111-DB/databases/w4111')
```

```
Connected to: ib://yy2827/W4111-DB/databases/w4111
```

In [2]:

```
%config SqlMagic.autocommit=False
ib.connect_db("postgresql://student:w4111student@w4111.cisxo09blonu.us-east-1.rds.amazonaws.com/w4111")
```

```
Connected to: postgresql://student:w4111student@w4111.cisx
o09blonu.us-east-1.rds.amazonaws.com/w4111
```

risk_rank table

In [3]:

```
%%sql

drop table if exists risk_rank cascade;

create table risk_rank(
    rid text primary key,
    risk_level int not null,
    keyword text not null
);
```

```
Done.
Done.
```

Out[3]:

□

venmo users

In [4]:

```
%%sql

drop table if exists venmo_users cascade;

create table venmo_users(
    user_name text primary key,
    full_name text not null,
    photo_link text,
    fb_account text
);
```

```
Done.
Done.
```

Out[4]:

□

transaction_transfer

In [5]:

```
%%sql

drop table if exists transa_transf cascade;

create table transa_transf(
    payment_id int primary key,
    created_time date not null,
    pay_charge text not null check (pay_charge = 'pay' or pay_charge
 ='charge'),
    actor_name text not null references venmo_users(user_name),
    target_name text not null references venmo_users(user_name),
    message text not null
);
```

```
Done.
Done.
```

Out[5]:

□

likes

In [6]:

```
%%sql
drop table if exists likes cascade;

create table likes(
    payment_id int references transa_transf,
    user_name text references venmo_users,
    primary key(payment_id,user_name)
);
```

Done.
Done.

Out[6]:

□

comments

In [7]:

```
%%sql
drop table if exists comment, comments cascade;

create table comments(
    com_id int primary key,
    contents text not null
);

create table comment(
    com_id int references comments,
    payment_id int references transa_transf,
    user_name text references venmo_users,
    primary key(com_id, payment_id,user_name)
);
```

```
Done.
Done.
Done.
```

Out[7]:

□

Bank_pay (weak entity for Venmo users)

In [8]:

```
%%sql
drop table if exists bank_pay;

create table bank_pay(
    card_id int primary key,
    user_name text not null references venmo_users on delete cascade,
    credit_debit text not null check (credit_debit = 'credit' or cred
it_debit ='debit'),
    bank_name text not null
);
```

```
Done.
Done.
```

Out[8]:

□

Populate the tables

risk_rank

In [9]:

```
%%sql
insert into risk_rank VALUES
('0000', 0, ''),
('1000', 1, 'venmo+public+risk'),
('0100', 1, 'venmo+friends+risk'),
('0010', 1, 'venmo+facebook+risk'),
('0001', 1, 'venmo+message+risk'),
('1100', 2, 'public+venmo+friends+risk'),
('1010', 2, 'public+venmo+facebook+risk'),
('1001', 2, 'public+venmo+message+risk'),
('0110', 2, 'venmo+friends+facebook+risk'),
('0101', 2, 'venmo+friends+message+risk'),
('0011', 2, 'venmo+facebook+message+risk'),
('1110', 3, 'public+venmo+friends+facebook+risk'),
('0111', 3, 'venmo+friends+facebook+message+risk'),
('1011', 3, 'public+venmo+facebook+message+risk'),
('1101', 3, 'public+venmo+friends+message+risk'),
('1111', 4, 'public+venmo+friends+facebook+message+risk');
```

16 rows affected.

Out[9]:

□

In [10]:

```
%%sql
select * from risk_rank
limit 5;
```

5 rows affected.

Out[10]:

	rid	risk_level	keyword
0	0000	0	
1	1000	1	venmo+public+risk
2	0100	1	venmo+friends+risk
3	0010	1	venmo+facebook+risk
4	0001	1	venmo+message+risk

venmo_users

In [11]:

```
%%sql
insert into venmo_users VALUES
( 'christianromeroUP', 'Christian Romero', 'https://graph.facebook.co
m/v2.10/10205730053050604/picture?type=large', '10205730053050604'),
( 'Jeffrey-Kemper-1', 'Jeffrey Kemper',  'https://venmopics.appspot.c
om/u/v1/m/c50b130d-feca-4dac-8caa-0961294bdd36', ''),
(  'Guo-Zhiqi', 'Guo Zhiqi', 'https://venmopics.appspot.com/u/v1/n/00
74e266-68ff-461b-96b5-9bb19661ec06', ''),
(  'MeganResnick', 'Megan Resnick',  'https://venmopics.appspot.com/
u/v3/n/e9692dd2-5a9f-4b1d-99b8-b074144b82ee', ''),
(  'NoahGershwin', 'Noah Gershwin', 'https://venmopics.appspot.com/u/
v1/m/09298cd9-cfb7-46b2-a364-cf59d505c0a4', ''),
( 'Marisa-Chambers', 'Marisa Chambers', 'https://graph.facebook.com/v
2.10/10207315721424646/picture?type=large', '10207315721424646'),
(  'elisadangelo', 'Elisa Dangelo', 'https://graph.facebook.com/v2.1
0/742587752539110/picture?type=large', '742587752539110'),
( 'anthonynguyen1006', 'Anthony Nguyen', 'https://venmopics.appspot.c
om/u/v2/m/f4d64bee-fcf6-4d50-932d-5f0d77875727', ''),
(  'jjscarz', 'Julia Scarangella', 'https://venmopics.appspot.com/u/v
1/m/d9ccd086-8cf3-4a1e-99ae-9bd12c1e67ee', ''),
(  'Michael-Caguioa', 'Michael Caguioa',  'https://venmopics.appspot.
com/u/v2/m/06b44f04-063a-4dc5-aa9d-410397fd023f', ''),
( 'Brian-Du-3', 'Brian Du', 'https://venmopics.appspot.com/u/v1/n/cba
5b23a-940d-4d2f-8df0-631d9a96f09e', ''),
( 'Andrew-Kwon-8', 'Andrew Kwon', 'https://venmopics.appspot.com/u/v
1/n/c66f063d-5e59-4ba3-a766-0e673323326d', '');
```

12 rows affected.

Out[11]:

□

In [12]:

```
%%sql
select * from venmo_users
```

12 rows affected.

Out[12]:

	user_name	full_name	
0	christianromeroUP	Christian Romero	https://graph.facebook.com/v2.10/102(
1	Jeffrey-Kemper-1	Jeffrey Kemper	https://venmopics.appspot.com/u/v1/m
2	Guo-Zhiqi	Guo Zhiqi	https://venmopics.appspot.com/u/v1/n
3	MeganResnick	Megan Resnick	https://venmopics.appspot.com/u/v3/n
4	NoahGershwin	Noah Gershwin	https://venmopics.appspot.com/u/v1/m
5	Marisa-Chambers	Marisa Chambers	https://graph.facebook.com/v2.10/102(
6	elisadangelo	Elisa Dangelo	https://graph.facebook.com/v2.10/742
7	anthonynguyen1006	Anthony Nguyen	https://venmopics.appspot.com/u/v2/m
8	jjscarz	Julia Scarangella	https://venmopics.appspot.com/u/v1/m
9	Michael-Caguioa	Michael Caguioa	https://venmopics.appspot.com/u/v2/m
10	Brian-Du-3	Brian Du	https://venmopics.appspot.com/u/v1/n
11	Andrew-Kwon-8	Andrew Kwon	https://venmopics.appspot.com/u/v1/n

transa_transf

In [13]:

```
%%sql
insert into transa_transf VALUES
(1239057877, '2018-10-22', 'pay', 'christianromeroUP', 'Jeffrey-Kempe
r-1', 'desk'),
(1239057876, '2018-10-22',  'pay', 'Jeffrey-Kemper-1', 'christianrome
roUP',  'Thanks bud'),
(1239057871, '2018-10-22', 'charge', 'Guo-Zhiqi', 'christianromeroU
P', 'Frothy monkey'),
(1239057870, '2018-10-21',  'pay', 'MeganResnick', 'christianromeroU
P', 'Chupando mis tetas'),
(1239057872, '2018-10-21', 'charge', 'christianromeroUP', 'NoahGershw
in', 'Love u ❤️☺️😩👌'),
(1239057865, '2018-10-21', 'charge', 'NoahGershwin', 'Marisa-Chamber
s', '🍕'),
(1239057863, '2018-10-21', 'pay', 'Marisa-Chambers', 'NoahGershwin',
 'Chipotle 🌯'),
(1239057859, '2018-10-20', 'charge', 'elisadangelo', 'NoahGershwin',
 '🍼'),
(1239057846, '2018-10-20', 'charge', 'NoahGershwin', 'anthonynguyen10
06', 'Uber from Nash airport to hotel'),
(1239057848, '2018-10-20',  'pay', 'jjscarz', 'NoahGershwin', 'Securi
ty deposit 💵⬅️BACK'),
(1239057849, '2018-10-20', 'charge', 'Michael-Caguioa', 'jjscarz','🍬
💁'),
(1239057850, '2018-10-20', 'pay', 'jjscarz', 'Brian-Du-3', 'Lyft+ dec
orations🌟'),
(1239057853, '2018-10-20', 'pay', 'Michael-Caguioa', 'jjscarz', 'Kara
oke'),
(1239057836, '2018-10-19', 'pay', 'jjscarz', 'Andrew-Kwon-8', 'U know
 what ;)'),
(1239057840, '2018-10-19', 'charge', 'Guo-Zhiqi', 'elisadangelo', 'Ti
to's'),
(1239057842, '2018-10-19', 'charge', 'Brian-Du-3','Guo-Zhiqi', 'Im do
ing hw at 3am and I just wanna go to my balcony and listen to sad mus
ic thnx for the groceries'),
(1239057832, '2018-10-19', 'pay', 'Guo-Zhiqi', 'jjscarz', 'Ur the bes
t');
```

17 rows affected.

Out[13]:

□

In [14]:

```
%%sql
select * from transa_transf
```

```
17 rows affected.
```

Out[14]:

	payment_id	created_time	pay_charge	actor_name	targ
0	1239057877	2018-10-22	pay	christianromeroUP	Jeffrey-Ker
1	1239057876	2018-10-22	pay	Jeffrey-Kemper-1	christianror
2	1239057871	2018-10-22	charge	Guo-Zhiqi	christianror
3	1239057870	2018-10-21	pay	MeganResnick	christianror
4	1239057872	2018-10-21	charge	christianromeroUP	NoahGersh
5	1239057865	2018-10-21	charge	NoahGershwin	Marisa-Cha
6	1239057863	2018-10-21	pay	Marisa-Chambers	NoahGersh
7	1239057859	2018-10-20	charge	elisadangelo	NoahGersh
8	1239057846	2018-10-20	charge	NoahGershwin	anthonyngu
9	1239057848	2018-10-20	pay	jjscarz	NoahGersh
10	1239057849	2018-10-20	charge	Michael-Caguioa	jjscarz
11	1239057850	2018-10-20	pay	jjscarz	Brian-Du-3
12	1239057853	2018-10-20	pay	Michael-Caguioa	jjscarz
13	1239057836	2018-10-19	pay	jjscarz	Andrew-Kw
14	1239057840	2018-10-19	charge	Guo-Zhiqi	elisadangel

	payment_id	created_time	pay_charge	actor_name	targ
15	1239057842	2018-10-19	charge	Brian-Du-3	Guo-Zhiqi
16	1239057832	2018-10-19	pay	Guo-Zhiqi	jjscarz

Bank_pay (weak entity for Venmo users)

In [15]:

```
%%sql
insert into bank_pay VALUES
(123971, 'christianromeroUP', 'credit', 'Chase'),
(123962, 'christianromeroUP',  'debit', 'BOA'),
(123913, 'Jeffrey-Kemper-1', 'credit', 'Citi'),
(123904, 'Guo-Zhiqi',  'credit', 'Discover'),
(123925, 'MeganResnick', 'debit', 'BNP'),
(123956, 'NoahGershwin', 'credit', 'RBC'),
(123937, 'Marisa-Chambers', 'credit', 'Chase'),
(123998, 'elisadangelo', 'debit', 'Citi'),
(123969, 'anthonynguyen1006', 'credit', 'BOA'),
(123980, 'jjscarz',  'credit', 'Chase'),
(123991, 'jjscarz', 'debit', 'DB'),
(123902, 'Michael-Caguioa', 'debit', 'Chase'),
(123933, 'Brian-Du-3', 'credit', 'HSBC'),
(123964, 'Andrew-Kwon-8', 'credit', 'Discover');
```

```
14 rows affected.
```

Out[15]:

□

In [16]:

```
%%sql
select * from bank_pay;
```

14 rows affected.

Out[16]:

	card_id	user_name	credit_debit	bank_name
0	123971	christianromeroUP	credit	Chase
1	123962	christianromeroUP	debit	BOA
2	123913	Jeffrey-Kemper-1	credit	Citi
3	123904	Guo-Zhiqi	credit	Discover
4	123925	MeganResnick	debit	BNP
5	123956	NoahGershwin	credit	RBC
6	123937	Marisa-Chambers	credit	Chase
7	123998	elisadangelo	debit	Citi
8	123969	anthonynguyen1006	credit	BOA
9	123980	jjscarz	credit	Chase
10	123991	jjscarz	debit	DB
11	123902	Michael-Caguioa	debit	Chase
12	123933	Brian-Du-3	credit	HSBC
13	123964	Andrew-Kwon-8	credit	Discover

likes

In [17]:

```
%%sql
insert into likes(payment_id,user_name) values
(1239057877,'christianromeroUP'),
(1239057877,'jjscarz'),
(1239057877,'Brian-Du-3'),
(1239057876,'Brian-Du-3'),
(1239057876,'Guo-Zhiqi'),
(1239057876,'NoahGershwin'),
(1239057876,'anthonynguyen1006'),
(1239057876,'christianromeroUP'),
(1239057876,'jjscarz'),
(1239057876,'Andrew-Kwon-8'),
(1239057876,'Marisa-Chambers'),
(1239057871,'Marisa-Chambers'),
(1239057871,'Jeffrey-Kemper-1'),
(1239057870,'Guo-Zhiqi'),
(1239057872,'Jeffrey-Kemper-1'),
(1239057872,'elisadangelo'),
(1239057865,'MeganResnick'),
(1239057865,'Brian-Du-3'),
(1239057865,'Marisa-Chambers'),
(1239057865,'Michael-Caguioa'),
(1239057863,'Jeffrey-Kemper-1'),
(1239057863,'christianromeroUP'),
(1239057863,'MeganResnick'),
(1239057863,'elisadangelo'),
(1239057846,'Jeffrey-Kemper-1'),
(1239057848,'MeganResnick'),
(1239057848,'anthonynguyen1006'),
(1239057850,'Guo-Zhiqi'),
(1239057850,'NoahGershwin'),
(1239057850,'anthonynguyen1006'),
(1239057853,'Michael-Caguioa'),
(1239057853,'jjscarz'),
(1239057853,'elisadangelo'),
(1239057853,'Jeffrey-Kemper-1'),
(1239057836,'Andrew-Kwon-8'),
(1239057836,'MeganResnick'),
(1239057840,'Andrew-Kwon-8'),
(1239057840,'NoahGershwin'),
(1239057842,'elisadangelo'),
(1239057842,'christianromeroUP'),
(1239057832,'Michael-Caguioa')
;
```

41 rows affected.

Out[17]:

□

In [18]:

```
%%sql
select distinct(payment_id) from likes;
```

15 rows affected.

Out[18]:

	payment_id
0	1239057840
1	1239057872
2	1239057863
3	1239057871
4	1239057850
5	1239057836
6	1239057848
7	1239057870
8	1239057876
9	1239057832
10	1239057865
11	1239057877
12	1239057846
13	1239057842
14	1239057853

comments

In [19]:

```
%%sql
insert into comments values
(1,'hahaha'),
(2,'cute'),
(3,'best'),
(4,'what?'),
(5,'so hardworking'),
(6,'omg take care'),
(7,'decorations for xmas?'),
(8,'sounds awsome'),
(9,'oh which one?'),
(10,'love chipotle'),
(11,'seems delicious'),
(12,'wow'),
(13,'Travel lol'),
(14,'love uuuuuu'),
(15,'no problem')
 ;
```

15 rows affected.

Out[19]:

□

In [20]:

```
%sql select * from comments
```

15 rows affected.

Out[20]:

	com_id	contents
0	1	hahaha
1	2	cute
2	3	best
3	4	what?
4	5	so hardworking
5	6	omg take care
6	7	decorations for xmas?
7	8	sounds awsome
8	9	oh which one?
9	10	love chipotle
10	11	seems delicious
11	12	wow
12	13	Travel lol
13	14	love uuuuuu
14	15	no problem

comment

In [21]:

```
%%sql
insert into comment values
(3,1239057832,'MeganResnick'),
(4,1239057836,'Brian-Du-3'),
(5,1239057842,'Guo-Zhiqi'),
(6,1239057842,'Andrew-Kwon-8'),
(1,1239057870,'christianromeroUP'),
(2,1239057849,'jjscarz'),
(7,1239057850,'Brian-Du-3'),
(8,1239057853,'Michael-Caguioa'),
(9,1239057853,'anthonynguyen1006'),
(2,1239057859,'elisadangelo'),
(10,1239057863,'Jeffrey-Kemper-1'),
(11,1239057865,'Marisa-Chambers'),
(12,1239057846,'jjscarz'),
(13,1239057846,'NoahGershwin'),
(1,1239057846,'jjscarz'),
(14,1239057872,'NoahGershwin'),
(15,1239057876,'christianromeroUP'),
(1,1239057876,'MeganResnick')
;
```

18 rows affected.

Out[21]:

□

In [22]:

```
%sql select * from comment;
```

18 rows affected.

Out[22]:

	com_id	payment_id	user_name
0	3	1239057832	MeganResnick
1	4	1239057836	Brian-Du-3
2	5	1239057842	Guo-Zhiqi
3	6	1239057842	Andrew-Kwon-8
4	1	1239057870	christianromeroUP
5	2	1239057849	jjscarz
6	7	1239057850	Brian-Du-3
7	8	1239057853	Michael-Caguioa
8	9	1239057853	anthonynguyen1006
9	2	1239057859	elisadangelo
10	10	1239057863	Jeffrey-Kemper-1
11	11	1239057865	Marisa-Chambers
12	12	1239057846	jjscarz
13	13	1239057846	NoahGershwin
14	1	1239057846	jjscarz
15	14	1239057872	NoahGershwin
16	15	1239057876	christianromeroUP
17	1	1239057876	MeganResnick

Queries

1. How many transactions per person in the past week?

In [23]:

```
%%sql
drop table if exists trans_no cascade;
create table trans_no as(
    with trans (username, name, trans_id) as (
        select tt.actor_name, vu.full_name, tt.payment_id from venmo_
users as vu
        join transa_transf as tt on vu.user_name = tt.actor_name
        where tt.created_time >= date'2018-10-22' - interval '7 days'
        union
        select tt.target_name, vu.full_name, tt.payment_id from venmo
_users as vu
        join transa_transf as tt on vu.user_name = tt.target_name
        where tt.created_time >= date'2018-10-22' - interval '7 days'
    )
    select username, name, count(*) as transactions from trans
    group by username, name
    order by transactions DESC
);
```

```
Done.
12 rows affected.
```

Out[23]:

In [24]:

```
%%sql
select * from trans_no;
```

12 rows affected.

Out[24]:

	username	name	transactions
0	NoahGershwin	Noah Gershwin	6
1	jjscarz	Julia Scarangella	6
2	christianromeroUP	Christian Romero	5
3	Guo-Zhiqi	Guo Zhiqi	4
4	Marisa-Chambers	Marisa Chambers	2
5	Michael-Caguioa	Michael Caguioa	2
6	elisadangelo	Elisa Dangelo	2
7	Brian-Du-3	Brian Du	2
8	Jeffrey-Kemper-1	Jeffrey Kemper	2
9	Andrew-Kwon-8	Andrew Kwon	1
10	MeganResnick	Megan Resnick	1
11	anthonynguyen1006	Anthony Nguyen	1

2. How many distinct people one user transacted with in the past week?

In [25]:

```
%%sql
with trans_person (username1, username2) as (
    select tt.actor_name, tt.target_name from transa_transf as tt
    where tt.created_time >= date'2018-10-22' - interval '7 days'
    union
    select tt.target_name, tt.actor_name from transa_transf as tt
    where tt.created_time >= date'2018-10-22' - interval '7 days'
)
select username1, vu.full_name as name, count(username1) as persons f
rom trans_person as tp
join venmo_users as vu on vu.user_name = tp.username1
group by username1, name
order by persons DESC;
```

12 rows affected.

Out[25]:

	username1	name	persons
0	NoahGershwin	Noah Gershwin	5
1	jjscarz	Julia Scarangella	5
2	christianromeroUP	Christian Romero	4
3	Guo-Zhiqi	Guo Zhiqi	4
4	elisadangelo	Elisa Dangelo	2
5	Brian-Du-3	Brian Du	2
6	Marisa-Chambers	Marisa Chambers	1
7	Andrew-Kwon-8	Andrew Kwon	1
8	anthonynguyen1006	Anthony Nguyen	1
9	Michael-Caguioa	Michael Caguioa	1
10	Jeffrey-Kemper-1	Jeffrey Kemper	1
11	MeganResnick	Megan Resnick	1

3. The largest length of message for each Venmo user's transactions

In [26]:

```
%%sql
drop table if exists mes cascade;
create table mes (username, name, trans_id, message) as (
    select tt.actor_name, vu.full_name, tt.payment_id, tt.message fro
m venmo_users as vu
    join transa_transf as tt on vu.user_name = tt.actor_name
    where tt.created_time >= date'2018-10-22' - interval '7 days'
);
drop function if exists strlen cascade;
CREATE FUNCTION strlen(mes) RETURNS integer AS $$
    SELECT char_length($1.message);
$$ LANGUAGE SQL;
select username, name, max(strlen(mes.*)) as length from mes
group by username, name
order by length DESC;
```

```
Done.
17 rows affected.
Done.
Done.
10 rows affected.
```

Out[26]:

	username	name	length
0	Brian-Du-3	Brian Du	99
1	NoahGershwin	Noah Gershwin	31
2	jjscarz	Julia Scarangella	19
3	MeganResnick	Megan Resnick	18
4	christianromeroUP	Christian Romero	14
5	Guo-Zhiqi	Guo Zhiqi	13
6	Jeffrey-Kemper-1	Jeffrey Kemper	10
7	Marisa-Chambers	Marisa Chambers	10
8	Michael-Caguioa	Michael Caguioa	7
9	elisadangelo	Elisa Dangelo	1

Part 3

1. create views for risk factors

(1). public transactions

In [27]:

```
%%sql
-- trx_history
drop view if exists trx_hist cascade;
create view trx_hist as (
  select vu.user_name, payment_id from venmo_users as vu
  join transa_transf as tt1
  on vu.user_name = tt1.actor_name
  union
  select vu.user_name, payment_id from venmo_users as vu
  join transa_transf as tt2
  on vu.user_name = tt2.target_name
);
```

```
Done.
Done.
```

Out[27]:

In [28]:

```
%sql select * from trx_hist order by user_name;
```

```
34 rows affected.
```

Out[28]:

	user_name	payment_id
0	Andrew-Kwon-8	1239057836
1	anthonynguyen1006	1239057846
2	Brian-Du-3	1239057850
3	Brian-Du-3	1239057842
4	christianromeroUP	1239057872
5	christianromeroUP	1239057870
6	christianromeroUP	1239057871
7	christianromeroUP	1239057876
8	christianromeroUP	1239057877
9	elisadangelo	1239057840
10	elisadangelo	1239057859
11	Guo-Zhiqi	1239057842
12	Guo-Zhiqi	1239057871
13	Guo-Zhiqi	1239057832
14	Guo-Zhiqi	1239057840
15	Jeffrey-Kemper-1	1239057876
16	Jeffrey-Kemper-1	1239057877
17	jjscarz	1239057848
18	jjscarz	1239057832
19	jjscarz	1239057849
20	jjscarz	1239057853
21	jjscarz	1239057836
22	jjscarz	1239057850
23	Marisa-Chambers	1239057865
24	Marisa-Chambers	1239057863
25	MeganResnick	1239057870

	user_name	payment_id
26	Michael-Caguioa	1239057853
27	Michael-Caguioa	1239057849
28	NoahGershwin	1239057848
29	NoahGershwin	1239057872
30	NoahGershwin	1239057863
31	NoahGershwin	1239057859
32	NoahGershwin	1239057846
33	NoahGershwin	1239057865

In [29]:

```
%%sql
-- personal_trx_count
drop view if exists person_trx cascade;
create view person_trx as (
  with temp as (
    select user_name, count(payment_id) from trx_hist
    group by user_name
    order by count(payment_id) desc
  )
  select *,
       case when count>= 5 then 1
            else 0
       end as indicator1
  from temp
);
```

Done.
Done.

Out[29]:

□

In [30]:

```
%sql select * from person_trx order by user_name;
```

12 rows affected.

Out[30]:

	user_name	count	indicator1
0	Andrew-Kwon-8	1	0
1	anthonynguyen1006	1	0
2	Brian-Du-3	2	0
3	christianromeroUP	5	1
4	elisadangelo	2	0
5	Guo-Zhiqi	4	0
6	Jeffrey-Kemper-1	2	0
7	jjscarz	6	1
8	Marisa-Chambers	2	0
9	MeganResnick	1	0
10	Michael-Caguioa	2	0
11	NoahGershwin	6	1

(2). friends

In [31]:

```
%%sql
-- calculate friends from likes, comments and transactions
drop view if exists friends cascade;

create view friends as (
  with temp as (
    select user_name, count(friends) as count from (
      select t.user_name as user_name, l.user_name as friends from tr
x_hist as t
      left join likes as l on t.payment_id = l.payment_id
      union
      select t.user_name as user_name, c.user_name as friends from tr
x_hist as t
      left join comment as c on t.payment_id = c.payment_id
      union
      select t1.user_name as user_name, t2.user_name as friends from
 trx_hist as t1
      left join trx_hist as t2 on t1.payment_id = t2.payment_id) as f
    where friends not in ('None',user_name)
    group by user_name
  )
  select vu.user_name, count,
         case when count>=3 then 1
              else 0
         end as indicator2
  from venmo_users as vu
  left join temp on vu.user_name = temp.user_name
);
```

Done.
Done.

Out[31]:

□

In [32]:

```
%%sql
select * from friends order by user_name;
```

12 rows affected.

Out[32]:

	user_name	count	indicator2
0	Andrew-Kwon-8	3	1
1	anthonynguyen1006	3	1
2	Brian-Du-3	7	1
3	christianromeroUP	10	1
4	elisadangelo	3	1
5	Guo-Zhiqi	10	1
6	Jeffrey-Kemper-1	9	1
7	jjscarz	9	1
8	Marisa-Chambers	7	1
9	MeganResnick	2	0
10	Michael-Caguioa	4	1
11	NoahGershwin	9	1

(3). link to FB

In [33]:

```
%%sql
drop view if exists fb cascade;
create view fb as (
  select user_name, fb_account,
         case when fb_account='' then 0
              else 1
         end as indicator3
  from venmo_users
);
```

Done.
Done.

Out[33]:

□

In [34]:

```
%sql select * from fb order by user_name;
```

12 rows affected.

Out[34]:

	user_name	fb_account	indicator3
0	Andrew-Kwon-8		0
1	anthonynguyen1006		0
2	Brian-Du-3		0
3	christianromeroUP	10205730053050604	1
4	elisadangelo	742587752539110	1
5	Guo-Zhiqi		0
6	Jeffrey-Kemper-1		0
7	jjscarz		0
8	Marisa-Chambers	10207315721424646	1
9	MeganResnick		0
10	Michael-Caguioa		0
11	NoahGershwin		0

(4). message length

In [35]:

```
%%sql
drop view if exists person_notes cascade;
create view person_notes as (
  select vu.user_name, max(length(tt.message)),
         case when max(length(tt.message))>=10 then 1
              else 0
         end as indicator4
  from venmo_users as vu
  left join trx_hist on vu.user_name = trx_hist.user_name
  left join transa_transf  as tt on trx_hist.payment_id = tt.payment_
id
  group by vu.user_name
);
```

```
Done.
Done.
```

Out[35]:

□

In [36]:

```
%sql select * from person_notes order by user_name;
```

12 rows affected.

Out[36]:

	user_name	max	indicator4
0	Andrew-Kwon-8	14	1
1	anthonynguyen1006	31	1
2	Brian-Du-3	99	1
3	christianromeroUP	18	1
4	elisadangelo	6	0
5	Guo-Zhiqi	99	1
6	Jeffrey-Kemper-1	10	1
7	jjscarz	19	1
8	Marisa-Chambers	10	1
9	MeganResnick	18	1
10	Michael-Caguioa	7	0
11	NoahGershwin	31	1

2. compute the personal risk evaluation result

In [37]:

```
%%sql

drop view if exists person_risk_result cascade;

create view person_risk_result as (
  select vu.user_name,
    CASE WHEN (indicator1 = 1) THEN 'true'
        ELSE 'false'
    END as too_many_payments,
    CASE WHEN (indicator2 = 1) THEN 'true'
        ELSE 'false'
    END as too_many_friends_shown,
    CASE WHEN (indicator3 = 1) THEN 'true'
        ELSE 'false'
    END as Track_to_Facebook,
    CASE WHEN (indicator4 = 1) THEN 'true'
        ELSE 'false'
    END as Too_detailed_notes,
         concat(indicator1,indicator2,indicator3,indicator4) as rid,
         risk_level,
         keyword
  from venmo_users as vu, person_trx as pt, friends as f,
       fb, person_notes as pn, risk_rank as rr
  where vu.user_name = pt.user_name
  and pt.user_name = f.user_name
  and f.user_name = fb.user_name
  and fb.user_name = pn.user_name
  and concat(indicator1,indicator2,indicator3,indicator4) = rr.rid
);
```

Done.
Done.

Out[37]:

□

In [38]:

```
%%sql
select * from person_risk_result;
```

12 rows affected.

Out[38]:

	user_name	too_many_payments	too_many_friends_shown	
0	christianromeroUP	true	true	
1	Jeffrey-Kemper-1	false	true	
2	Guo-Zhiqi	false	true	
3	MeganResnick	false	false	
4	NoahGershwin	true	true	
5	Marisa-Chambers	false	true	
6	elisadangelo	false	true	
7	anthonynguyen1006	false	true	
8	jjscarz	true	true	
9	Michael-Caguioa	false	true	
10	Brian-Du-3	false	true	
11	Andrew-Kwon-8	false	true	

In [1]:

```
ib.connect_db('ib://yy2827/W4111-DB/databases/w4111')
```

```
Connected to: ib://yy2827/W4111-DB/databases/w4111
```

In [2]:

```
%config SqlMagic.autocommit=False
ib.connect_db("postgresql://student:w4111student@w4111.cisxo09blonu.us-east-1.rds.amazonaws.com/w4111")
```

```
Connected to: postgresql://student:w4111student@w4111.cisx
o09blonu.us-east-1.rds.amazonaws.com/w4111
```

risk_rank table

In [3]:

```
%%sql

drop table if exists risk_rank cascade;

create table risk_rank(
    rid text primary key,
    risk_level int not null,
    keyword text not null
);
```

```
Done.
Done.
```

Out[3]:

□

venmo users

In [4]:

```
%%sql

drop table if exists venmo_users cascade;

create table venmo_users(
    user_name text primary key,
    full_name text not null,
    photo_link text,
    fb_account text
);
```

```
Done.
Done.
```

Out[4]:

□

transaction_transfer

In [5]:

```
%%sql

drop table if exists transa_transf cascade;

create table transa_transf(
    payment_id int primary key,
    created_time date not null,
    pay_charge text not null check (pay_charge = 'pay' or pay_charge
 ='charge'),
    actor_name text not null references venmo_users(user_name),
    target_name text not null references venmo_users(user_name),
    message text not null
);
```

```
Done.
Done.
```

Out[5]:

□

likes

In [6]:

```
%%sql
drop table if exists likes cascade;

create table likes(
    payment_id int references transa_transf,
    user_name text references venmo_users,
    primary key(payment_id,user_name)
);
```

Done.
Done.

Out[6]:

□

comments

In [7]:

```
%%sql
drop table if exists comment, comments cascade;

create table comments(
    com_id int primary key,
    contents text not null
);

create table comment(
    com_id int references comments,
    payment_id int references transa_transf,
    user_name text references venmo_users,
    primary key(com_id, payment_id,user_name)
);
```

```
Done.
Done.
Done.
```

Out[7]:

□

Bank_pay (weak entity for Venmo users)

In [8]:

```
%%sql
drop table if exists bank_pay;

create table bank_pay(
    card_id int primary key,
    user_name text not null references venmo_users on delete cascade,
    credit_debit text not null check (credit_debit = 'credit' or cred
it_debit ='debit'),
    bank_name text not null
);
```

```
Done.
Done.
```

Out[8]:

□

Populate the tables

risk_rank

In [9]:

```
%%sql
insert into risk_rank VALUES
('0000', 0, ''),
('1000', 1, 'venmo+public+risk'),
('0100', 1, 'venmo+friends+risk'),
('0010', 1, 'venmo+facebook+risk'),
('0001', 1, 'venmo+message+risk'),
('1100', 2, 'public+venmo+friends+risk'),
('1010', 2, 'public+venmo+facebook+risk'),
('1001', 2, 'public+venmo+message+risk'),
('0110', 2, 'venmo+friends+facebook+risk'),
('0101', 2, 'venmo+friends+message+risk'),
('0011', 2, 'venmo+facebook+message+risk'),
('1110', 3, 'public+venmo+friends+facebook+risk'),
('0111', 3, 'venmo+friends+facebook+message+risk'),
('1011', 3, 'public+venmo+facebook+message+risk'),
('1101', 3, 'public+venmo+friends+message+risk'),
('1111', 4, 'public+venmo+friends+facebook+message+risk');
```

16 rows affected.

Out[9]:

□

In [10]:

```
%%sql
select * from risk_rank
limit 5;
```

5 rows affected.

Out[10]:

	rid	risk_level	keyword
0	0000	0	
1	1000	1	venmo+public+risk
2	0100	1	venmo+friends+risk
3	0010	1	venmo+facebook+risk
4	0001	1	venmo+message+risk

venmo_users

In [11]:

```
%%sql
insert into venmo_users VALUES
( 'christianromeroUP', 'Christian Romero', 'https://graph.facebook.co
m/v2.10/10205730053050604/picture?type=large', '10205730053050604'),
( 'Jeffrey-Kemper-1', 'Jeffrey Kemper',  'https://venmopics.appspot.c
om/u/v1/m/c50b130d-feca-4dac-8caa-0961294bdd36', ''),
(  'Guo-Zhiqi', 'Guo Zhiqi', 'https://venmopics.appspot.com/u/v1/n/00
74e266-68ff-461b-96b5-9bb19661ec06', ''),
(  'MeganResnick', 'Megan Resnick',  'https://venmopics.appspot.com/
u/v3/n/e9692dd2-5a9f-4b1d-99b8-b074144b82ee', ''),
(  'NoahGershwin', 'Noah Gershwin', 'https://venmopics.appspot.com/u/
v1/m/09298cd9-cfb7-46b2-a364-cf59d505c0a4', ''),
( 'Marisa-Chambers', 'Marisa Chambers', 'https://graph.facebook.com/v
2.10/10207315721424646/picture?type=large', '10207315721424646'),
(  'elisadangelo', 'Elisa Dangelo', 'https://graph.facebook.com/v2.1
0/742587752539110/picture?type=large', '742587752539110'),
( 'anthonynguyen1006', 'Anthony Nguyen', 'https://venmopics.appspot.c
om/u/v2/m/f4d64bee-fcf6-4d50-932d-5f0d77875727', ''),
(  'jjscarz', 'Julia Scarangella', 'https://venmopics.appspot.com/u/v
1/m/d9ccd086-8cf3-4a1e-99ae-9bd12c1e67ee', ''),
(  'Michael-Caguioa', 'Michael Caguioa',  'https://venmopics.appspot.
com/u/v2/m/06b44f04-063a-4dc5-aa9d-410397fd023f', ''),
( 'Brian-Du-3', 'Brian Du', 'https://venmopics.appspot.com/u/v1/n/cba
5b23a-940d-4d2f-8df0-631d9a96f09e', ''),
( 'Andrew-Kwon-8', 'Andrew Kwon', 'https://venmopics.appspot.com/u/v
1/n/c66f063d-5e59-4ba3-a766-0e673323326d', '');
```

12 rows affected.

Out[11]:

□

In [12]:

```
%%sql
select * from venmo_users
```

12 rows affected.

Out[12]:

	user_name	full_name	
0	christianromeroUP	Christian Romero	https://graph.facebook.com/v2.10/102(
1	Jeffrey-Kemper-1	Jeffrey Kemper	https://venmopics.appspot.com/u/v1/m
2	Guo-Zhiqi	Guo Zhiqi	https://venmopics.appspot.com/u/v1/n
3	MeganResnick	Megan Resnick	https://venmopics.appspot.com/u/v3/n
4	NoahGershwin	Noah Gershwin	https://venmopics.appspot.com/u/v1/m
5	Marisa-Chambers	Marisa Chambers	https://graph.facebook.com/v2.10/102(
6	elisadangelo	Elisa Dangelo	https://graph.facebook.com/v2.10/742
7	anthonynguyen1006	Anthony Nguyen	https://venmopics.appspot.com/u/v2/m
8	jjscarz	Julia Scarangella	https://venmopics.appspot.com/u/v1/m
9	Michael-Caguioa	Michael Caguioa	https://venmopics.appspot.com/u/v2/m
10	Brian-Du-3	Brian Du	https://venmopics.appspot.com/u/v1/n
11	Andrew-Kwon-8	Andrew Kwon	https://venmopics.appspot.com/u/v1/n

transa_transf

In [13]:

```
%%sql
insert into transa_transf VALUES
(1239057877, '2018-10-22', 'pay', 'christianromeroUP', 'Jeffrey-Kempe
r-1', 'desk'),
(1239057876, '2018-10-22',  'pay', 'Jeffrey-Kemper-1', 'christianrome
roUP',  'Thanks bud'),
(1239057871, '2018-10-22', 'charge', 'Guo-Zhiqi', 'christianromeroU
P', 'Frothy monkey'),
(1239057870, '2018-10-21',  'pay', 'MeganResnick', 'christianromeroU
P', 'Chupando mis tetas'),
(1239057872, '2018-10-21', 'charge', 'christianromeroUP', 'NoahGershw
in', 'Love u ❤️☺️😩👌'),
(1239057865, '2018-10-21', 'charge', 'NoahGershwin', 'Marisa-Chamber
s', '🍕'),
(1239057863, '2018-10-21', 'pay', 'Marisa-Chambers', 'NoahGershwin',
 'Chipotle 🌯'),
(1239057859, '2018-10-20', 'charge', 'elisadangelo', 'NoahGershwin',
 '🍼'),
(1239057846, '2018-10-20', 'charge', 'NoahGershwin', 'anthonynguyen10
06', 'Uber from Nash airport to hotel'),
(1239057848, '2018-10-20',  'pay', 'jjscarz', 'NoahGershwin', 'Securi
ty deposit 💵⬅️BACK'),
(1239057849, '2018-10-20', 'charge', 'Michael-Caguioa', 'jjscarz','🍬
💁'),
(1239057850, '2018-10-20', 'pay', 'jjscarz', 'Brian-Du-3', 'Lyft+ dec
orations🌟'),
(1239057853, '2018-10-20', 'pay', 'Michael-Caguioa', 'jjscarz', 'Kara
oke'),
(1239057836, '2018-10-19', 'pay', 'jjscarz', 'Andrew-Kwon-8', 'U know
 what ;)'),
(1239057840, '2018-10-19', 'charge', 'Guo-Zhiqi', 'elisadangelo', 'Ti
to's'),
(1239057842, '2018-10-19', 'charge', 'Brian-Du-3','Guo-Zhiqi', 'Im do
ing hw at 3am and I just wanna go to my balcony and listen to sad mus
ic thnx for the groceries'),
(1239057832, '2018-10-19', 'pay', 'Guo-Zhiqi', 'jjscarz', 'Ur the bes
t');
```

17 rows affected.

Out[13]:

□

In [14]:

```
%%sql
select * from transa_transf
```

```
17 rows affected.
```

Out[14]:

	payment_id	created_time	pay_charge	actor_name	targ
0	1239057877	2018-10-22	pay	christianromeroUP	Jeffrey-Ker
1	1239057876	2018-10-22	pay	Jeffrey-Kemper-1	christianror
2	1239057871	2018-10-22	charge	Guo-Zhiqi	christianror
3	1239057870	2018-10-21	pay	MeganResnick	christianror
4	1239057872	2018-10-21	charge	christianromeroUP	NoahGersh
5	1239057865	2018-10-21	charge	NoahGershwin	Marisa-Cha
6	1239057863	2018-10-21	pay	Marisa-Chambers	NoahGersh
7	1239057859	2018-10-20	charge	elisadangelo	NoahGersh
8	1239057846	2018-10-20	charge	NoahGershwin	anthonyngu
9	1239057848	2018-10-20	pay	jjscarz	NoahGersh
10	1239057849	2018-10-20	charge	Michael-Caguioa	jjscarz
11	1239057850	2018-10-20	pay	jjscarz	Brian-Du-3
12	1239057853	2018-10-20	pay	Michael-Caguioa	jjscarz
13	1239057836	2018-10-19	pay	jjscarz	Andrew-Kw
14	1239057840	2018-10-19	charge	Guo-Zhiqi	elisadangel

	payment_id	created_time	pay_charge	actor_name	targ
15	1239057842	2018-10-19	charge	Brian-Du-3	Guo-Zhiqi
16	1239057832	2018-10-19	pay	Guo-Zhiqi	jjscarz

Bank_pay (weak entity for Venmo users)

In [15]:

```
%%sql
insert into bank_pay VALUES
(123971, 'christianromeroUP', 'credit', 'Chase'),
(123962, 'christianromeroUP',  'debit', 'BOA'),
(123913, 'Jeffrey-Kemper-1', 'credit', 'Citi'),
(123904, 'Guo-Zhiqi',  'credit', 'Discover'),
(123925, 'MeganResnick', 'debit', 'BNP'),
(123956, 'NoahGershwin', 'credit', 'RBC'),
(123937, 'Marisa-Chambers', 'credit', 'Chase'),
(123998, 'elisadangelo', 'debit', 'Citi'),
(123969, 'anthonynguyen1006', 'credit', 'BOA'),
(123980, 'jjscarz',  'credit', 'Chase'),
(123991, 'jjscarz', 'debit', 'DB'),
(123902, 'Michael-Caguioa', 'debit', 'Chase'),
(123933, 'Brian-Du-3', 'credit', 'HSBC'),
(123964, 'Andrew-Kwon-8', 'credit', 'Discover');
```

```
14 rows affected.
```

Out[15]:

□

In [16]:

```
%%sql
select * from bank_pay;
```

14 rows affected.

Out[16]:

	card_id	user_name	credit_debit	bank_name
0	123971	christianromeroUP	credit	Chase
1	123962	christianromeroUP	debit	BOA
2	123913	Jeffrey-Kemper-1	credit	Citi
3	123904	Guo-Zhiqi	credit	Discover
4	123925	MeganResnick	debit	BNP
5	123956	NoahGershwin	credit	RBC
6	123937	Marisa-Chambers	credit	Chase
7	123998	elisadangelo	debit	Citi
8	123969	anthonynguyen1006	credit	BOA
9	123980	jjscarz	credit	Chase
10	123991	jjscarz	debit	DB
11	123902	Michael-Caguioa	debit	Chase
12	123933	Brian-Du-3	credit	HSBC
13	123964	Andrew-Kwon-8	credit	Discover

likes

In [17]:

```
%%sql
insert into likes(payment_id,user_name) values
(1239057877,'christianromeroUP'),
(1239057877,'jjscarz'),
(1239057877,'Brian-Du-3'),
(1239057876,'Brian-Du-3'),
(1239057876,'Guo-Zhiqi'),
(1239057876,'NoahGershwin'),
(1239057876,'anthonynguyen1006'),
(1239057876,'christianromeroUP'),
(1239057876,'jjscarz'),
(1239057876,'Andrew-Kwon-8'),
(1239057876,'Marisa-Chambers'),
(1239057871,'Marisa-Chambers'),
(1239057871,'Jeffrey-Kemper-1'),
(1239057870,'Guo-Zhiqi'),
(1239057872,'Jeffrey-Kemper-1'),
(1239057872,'elisadangelo'),
(1239057865,'MeganResnick'),
(1239057865,'Brian-Du-3'),
(1239057865,'Marisa-Chambers'),
(1239057865,'Michael-Caguioa'),
(1239057863,'Jeffrey-Kemper-1'),
(1239057863,'christianromeroUP'),
(1239057863,'MeganResnick'),
(1239057863,'elisadangelo'),
(1239057846,'Jeffrey-Kemper-1'),
(1239057848,'MeganResnick'),
(1239057848,'anthonynguyen1006'),
(1239057850,'Guo-Zhiqi'),
(1239057850,'NoahGershwin'),
(1239057850,'anthonynguyen1006'),
(1239057853,'Michael-Caguioa'),
(1239057853,'jjscarz'),
(1239057853,'elisadangelo'),
(1239057853,'Jeffrey-Kemper-1'),
(1239057836,'Andrew-Kwon-8'),
(1239057836,'MeganResnick'),
(1239057840,'Andrew-Kwon-8'),
(1239057840,'NoahGershwin'),
(1239057842,'elisadangelo'),
(1239057842,'christianromeroUP'),
(1239057832,'Michael-Caguioa')
;
```

41 rows affected.

Out[17]:

□

In [18]:

```
%%sql
select distinct(payment_id) from likes;
```

15 rows affected.

Out[18]:

	payment_id
0	1239057840
1	1239057872
2	1239057863
3	1239057871
4	1239057850
5	1239057836
6	1239057848
7	1239057870
8	1239057876
9	1239057832
10	1239057865
11	1239057877
12	1239057846
13	1239057842
14	1239057853

comments

In [19]:

```
%%sql
insert into comments values
(1,'hahaha'),
(2,'cute'),
(3,'best'),
(4,'what?'),
(5,'so hardworking'),
(6,'omg take care'),
(7,'decorations for xmas?'),
(8,'sounds awsome'),
(9,'oh which one?'),
(10,'love chipotle'),
(11,'seems delicious'),
(12,'wow'),
(13,'Travel lol'),
(14,'love uuuuuu'),
(15,'no problem')
 ;
```

15 rows affected.

Out[19]:

□

In [20]:

```
%sql select * from comments
```

15 rows affected.

Out[20]:

	com_id	contents
0	1	hahaha
1	2	cute
2	3	best
3	4	what?
4	5	so hardworking
5	6	omg take care
6	7	decorations for xmas?
7	8	sounds awsome
8	9	oh which one?
9	10	love chipotle
10	11	seems delicious
11	12	wow
12	13	Travel lol
13	14	love uuuuuu
14	15	no problem

comment

In [21]:

```
%%sql
insert into comment values
(3,1239057832,'MeganResnick'),
(4,1239057836,'Brian-Du-3'),
(5,1239057842,'Guo-Zhiqi'),
(6,1239057842,'Andrew-Kwon-8'),
(1,1239057870,'christianromeroUP'),
(2,1239057849,'jjscarz'),
(7,1239057850,'Brian-Du-3'),
(8,1239057853,'Michael-Caguioa'),
(9,1239057853,'anthonynguyen1006'),
(2,1239057859,'elisadangelo'),
(10,1239057863,'Jeffrey-Kemper-1'),
(11,1239057865,'Marisa-Chambers'),
(12,1239057846,'jjscarz'),
(13,1239057846,'NoahGershwin'),
(1,1239057846,'jjscarz'),
(14,1239057872,'NoahGershwin'),
(15,1239057876,'christianromeroUP'),
(1,1239057876,'MeganResnick')
;
```

18 rows affected.

Out[21]:

□

In [22]:

```
%sql select * from comment;
```

18 rows affected.

Out[22]:

	com_id	payment_id	user_name
0	3	1239057832	MeganResnick
1	4	1239057836	Brian-Du-3
2	5	1239057842	Guo-Zhiqi
3	6	1239057842	Andrew-Kwon-8
4	1	1239057870	christianromeroUP
5	2	1239057849	jjscarz
6	7	1239057850	Brian-Du-3
7	8	1239057853	Michael-Caguioa
8	9	1239057853	anthonynguyen1006
9	2	1239057859	elisadangelo
10	10	1239057863	Jeffrey-Kemper-1
11	11	1239057865	Marisa-Chambers
12	12	1239057846	jjscarz
13	13	1239057846	NoahGershwin
14	1	1239057846	jjscarz
15	14	1239057872	NoahGershwin
16	15	1239057876	christianromeroUP
17	1	1239057876	MeganResnick

Queries

1. How many transactions per person in the past week?

In [23]:

```
%%sql
drop table if exists trans_no cascade;
create table trans_no as(
    with trans (username, name, trans_id) as (
        select tt.actor_name, vu.full_name, tt.payment_id from venmo_
users as vu
        join transa_transf as tt on vu.user_name = tt.actor_name
        where tt.created_time >= date'2018-10-22' - interval '7 days'
        union
        select tt.target_name, vu.full_name, tt.payment_id from venmo
_users as vu
        join transa_transf as tt on vu.user_name = tt.target_name
        where tt.created_time >= date'2018-10-22' - interval '7 days'
    )
    select username, name, count(*) as transactions from trans
    group by username, name
    order by transactions DESC
);
```

```
Done.
12 rows affected.
```

Out[23]:

In [24]:

```
%%sql
select * from trans_no;
```

12 rows affected.

Out[24]:

	username	name	transactions
0	NoahGershwin	Noah Gershwin	6
1	jjscarz	Julia Scarangella	6
2	christianromeroUP	Christian Romero	5
3	Guo-Zhiqi	Guo Zhiqi	4
4	Marisa-Chambers	Marisa Chambers	2
5	Michael-Caguioa	Michael Caguioa	2
6	elisadangelo	Elisa Dangelo	2
7	Brian-Du-3	Brian Du	2
8	Jeffrey-Kemper-1	Jeffrey Kemper	2
9	Andrew-Kwon-8	Andrew Kwon	1
10	MeganResnick	Megan Resnick	1
11	anthonynguyen1006	Anthony Nguyen	1

2. How many distinct people one user transacted with in the past week?

In [25]:

```
%%sql
with trans_person (username1, username2) as (
    select tt.actor_name, tt.target_name from transa_transf as tt
    where tt.created_time >= date'2018-10-22' - interval '7 days'
    union
    select tt.target_name, tt.actor_name from transa_transf as tt
    where tt.created_time >= date'2018-10-22' - interval '7 days'
)
select username1, vu.full_name as name, count(username1) as persons f
rom trans_person as tp
join venmo_users as vu on vu.user_name = tp.username1
group by username1, name
order by persons DESC;
```

12 rows affected.

Out[25]:

	username1	name	persons
0	NoahGershwin	Noah Gershwin	5
1	jjscarz	Julia Scarangella	5
2	christianromeroUP	Christian Romero	4
3	Guo-Zhiqi	Guo Zhiqi	4
4	elisadangelo	Elisa Dangelo	2
5	Brian-Du-3	Brian Du	2
6	Marisa-Chambers	Marisa Chambers	1
7	Andrew-Kwon-8	Andrew Kwon	1
8	anthonynguyen1006	Anthony Nguyen	1
9	Michael-Caguioa	Michael Caguioa	1
10	Jeffrey-Kemper-1	Jeffrey Kemper	1
11	MeganResnick	Megan Resnick	1

3. The largest length of message for each Venmo user's transactions

In [26]:

```
%%sql
drop table if exists mes cascade;
create table mes (username, name, trans_id, message) as (
    select tt.actor_name, vu.full_name, tt.payment_id, tt.message fro
m venmo_users as vu
    join transa_transf as tt on vu.user_name = tt.actor_name
    where tt.created_time >= date'2018-10-22' - interval '7 days'
);
drop function if exists strlen cascade;
CREATE FUNCTION strlen(mes) RETURNS integer AS $$
    SELECT char_length($1.message);
$$ LANGUAGE SQL;
select username, name, max(strlen(mes.*)) as length from mes
group by username, name
order by length DESC;
```

```
Done.
17 rows affected.
Done.
Done.
10 rows affected.
```

Out[26]:

	username	name	length
0	Brian-Du-3	Brian Du	99
1	NoahGershwin	Noah Gershwin	31
2	jjscarz	Julia Scarangella	19
3	MeganResnick	Megan Resnick	18
4	christianromeroUP	Christian Romero	14
5	Guo-Zhiqi	Guo Zhiqi	13
6	Jeffrey-Kemper-1	Jeffrey Kemper	10
7	Marisa-Chambers	Marisa Chambers	10
8	Michael-Caguioa	Michael Caguioa	7
9	elisadangelo	Elisa Dangelo	1

Part 3

1. create views for risk factors

(1). public transactions

In [27]:

```
%%sql
-- trx_history
drop view if exists trx_hist cascade;
create view trx_hist as (
  select vu.user_name, payment_id from venmo_users as vu
  join transa_transf as tt1
  on vu.user_name = tt1.actor_name
  union
  select vu.user_name, payment_id from venmo_users as vu
  join transa_transf as tt2
  on vu.user_name = tt2.target_name
);
```

```
Done.
Done.
```

Out[27]:

In [28]:

```
%sql select * from trx_hist order by user_name;
```

```
34 rows affected.
```

Out[28]:

	user_name	payment_id
0	Andrew-Kwon-8	1239057836
1	anthonynguyen1006	1239057846
2	Brian-Du-3	1239057850
3	Brian-Du-3	1239057842
4	christianromeroUP	1239057872
5	christianromeroUP	1239057870
6	christianromeroUP	1239057871
7	christianromeroUP	1239057876
8	christianromeroUP	1239057877
9	elisadangelo	1239057840
10	elisadangelo	1239057859
11	Guo-Zhiqi	1239057842
12	Guo-Zhiqi	1239057871
13	Guo-Zhiqi	1239057832
14	Guo-Zhiqi	1239057840
15	Jeffrey-Kemper-1	1239057876
16	Jeffrey-Kemper-1	1239057877
17	jjscarz	1239057848
18	jjscarz	1239057832
19	jjscarz	1239057849
20	jjscarz	1239057853
21	jjscarz	1239057836
22	jjscarz	1239057850
23	Marisa-Chambers	1239057865
24	Marisa-Chambers	1239057863
25	MeganResnick	1239057870

	user_name	payment_id
26	Michael-Caguioa	1239057853
27	Michael-Caguioa	1239057849
28	NoahGershwin	1239057848
29	NoahGershwin	1239057872
30	NoahGershwin	1239057863
31	NoahGershwin	1239057859
32	NoahGershwin	1239057846
33	NoahGershwin	1239057865

In [29]:

```
%%sql
-- personal_trx_count
drop view if exists person_trx cascade;
create view person_trx as (
  with temp as (
    select user_name, count(payment_id) from trx_hist
    group by user_name
    order by count(payment_id) desc
  )
  select *,
       case when count>= 5 then 1
            else 0
       end as indicator1
  from temp
);
```

Done.
Done.

Out[29]:

□

In [30]:

```
%sql select * from person_trx order by user_name;
```

12 rows affected.

Out[30]:

	user_name	count	indicator1
0	Andrew-Kwon-8	1	0
1	anthonynguyen1006	1	0
2	Brian-Du-3	2	0
3	christianromeroUP	5	1
4	elisadangelo	2	0
5	Guo-Zhiqi	4	0
6	Jeffrey-Kemper-1	2	0
7	jjscarz	6	1
8	Marisa-Chambers	2	0
9	MeganResnick	1	0
10	Michael-Caguioa	2	0
11	NoahGershwin	6	1

(2). friends

In [31]:

```
%%sql
-- calculate friends from likes, comments and transactions
drop view if exists friends cascade;

create view friends as (
  with temp as (
    select user_name, count(friends) as count from (
      select t.user_name as user_name, l.user_name as friends from tr
x_hist as t
      left join likes as l on t.payment_id = l.payment_id
      union
      select t.user_name as user_name, c.user_name as friends from tr
x_hist as t
      left join comment as c on t.payment_id = c.payment_id
      union
      select t1.user_name as user_name, t2.user_name as friends from
 trx_hist as t1
      left join trx_hist as t2 on t1.payment_id = t2.payment_id) as f
    where friends not in ('None',user_name)
    group by user_name
  )
  select vu.user_name, count,
         case when count>=3 then 1
              else 0
         end as indicator2
  from venmo_users as vu
  left join temp on vu.user_name = temp.user_name
);
```

Done.
Done.

Out[31]:

□

In [32]:

```
%%sql
select * from friends order by user_name;
```

12 rows affected.

Out[32]:

	user_name	count	indicator2
0	Andrew-Kwon-8	3	1
1	anthonynguyen1006	3	1
2	Brian-Du-3	7	1
3	christianromeroUP	10	1
4	elisadangelo	3	1
5	Guo-Zhiqi	10	1
6	Jeffrey-Kemper-1	9	1
7	jjscarz	9	1
8	Marisa-Chambers	7	1
9	MeganResnick	2	0
10	Michael-Caguioa	4	1
11	NoahGershwin	9	1

(3). link to FB

In [33]:

```
%%sql
drop view if exists fb cascade;
create view fb as (
  select user_name, fb_account,
         case when fb_account='' then 0
              else 1
         end as indicator3
  from venmo_users
);
```

Done.
Done.

Out[33]:

□

In [34]:

```
%sql select * from fb order by user_name;
```

12 rows affected.

Out[34]:

	user_name	fb_account	indicator3
0	Andrew-Kwon-8		0
1	anthonynguyen1006		0
2	Brian-Du-3		0
3	christianromeroUP	10205730053050604	1
4	elisadangelo	742587752539110	1
5	Guo-Zhiqi		0
6	Jeffrey-Kemper-1		0
7	jjscarz		0
8	Marisa-Chambers	10207315721424646	1
9	MeganResnick		0
10	Michael-Caguioa		0
11	NoahGershwin		0

(4). message length

In [35]:

```
%%sql
drop view if exists person_notes cascade;
create view person_notes as (
  select vu.user_name, max(length(tt.message)),
         case when max(length(tt.message))>=10 then 1
              else 0
         end as indicator4
  from venmo_users as vu
  left join trx_hist on vu.user_name = trx_hist.user_name
  left join transa_transf  as tt on trx_hist.payment_id = tt.payment_
id
  group by vu.user_name
);
```

```
Done.
Done.
```

Out[35]:

□

In [36]:

```
%sql select * from person_notes order by user_name;
```

12 rows affected.

Out[36]:

	user_name	max	indicator4
0	Andrew-Kwon-8	14	1
1	anthonynguyen1006	31	1
2	Brian-Du-3	99	1
3	christianromeroUP	18	1
4	elisadangelo	6	0
5	Guo-Zhiqi	99	1
6	Jeffrey-Kemper-1	10	1
7	jjscarz	19	1
8	Marisa-Chambers	10	1
9	MeganResnick	18	1
10	Michael-Caguioa	7	0
11	NoahGershwin	31	1

2. compute the personal risk evaluation result

In [37]:

```
%%sql

drop view if exists person_risk_result cascade;

create view person_risk_result as (
  select vu.user_name,
    CASE WHEN (indicator1 = 1) THEN 'true'
        ELSE 'false'
    END as too_many_payments,
    CASE WHEN (indicator2 = 1) THEN 'true'
        ELSE 'false'
    END as too_many_friends_shown,
    CASE WHEN (indicator3 = 1) THEN 'true'
        ELSE 'false'
    END as Track_to_Facebook,
    CASE WHEN (indicator4 = 1) THEN 'true'
        ELSE 'false'
    END as Too_detailed_notes,
         concat(indicator1,indicator2,indicator3,indicator4) as rid,
         risk_level,
         keyword
  from venmo_users as vu, person_trx as pt, friends as f,
       fb, person_notes as pn, risk_rank as rr
  where vu.user_name = pt.user_name
  and pt.user_name = f.user_name
  and f.user_name = fb.user_name
  and fb.user_name = pn.user_name
  and concat(indicator1,indicator2,indicator3,indicator4) = rr.rid
);
```

Done.
Done.

Out[37]:

□

In [38]:

```
%%sql
select * from person_risk_result;
```

12 rows affected.

Out[38]:

	user_name	too_many_payments	too_many_friends_shown	
0	christianromeroUP	true	true	
1	Jeffrey-Kemper-1	false	true	
2	Guo-Zhiqi	false	true	
3	MeganResnick	false	false	
4	NoahGershwin	true	true	
5	Marisa-Chambers	false	true	
6	elisadangelo	false	true	
7	anthonynguyen1006	false	true	
8	jjscarz	true	true	
9	Michael-Caguioa	false	true	
10	Brian-Du-3	false	true	
11	Andrew-Kwon-8	false	true	