Slack Code Exercise Candidate: Dwight Liu

I create two tables email and recipient
Email table describe

```
| Field      | Type             | Null | Key | Default | Extra          |
+------------+------------------+------+-----+---------+----------------+
| id         | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| message_id | varchar(250)     | NO   | UNI |         |                |
| sender     | varchar(150)     | YES  |     | NULL    |                |
| subject    | varchar(400)     | YES  |     | NULL    |                |
| email_date | datetime         | YES  |     | NULL    |                |
| label      | varchar(15)      | YES  |     | NULL    |                |
| sub_md5    | varchar(55)      | YES  |     | NULL    |                |
+------------+------------------+------+-----+---------+----------------+
| Field      | Type             | Null | Key | Default | Extra          |
+------------+------------------+------+-----+---------+----------------+
| id         | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| message_id | varchar(250)     | NO   |     |         |                |
| sender     | varchar(150)     | YES  |     | NULL    |                |
| recipient  | varchar(150)     | YES  |     |         |                |
| subject    | varchar(400)     | YES  |     | NULL    |                |
| sub_md5    | varchar(55)      | YES  |     | NULL    |                |
| email_date | datetime         | YES  |     | NULL    |                |
| is_to      | tinyint(1)       | YES  |     | NULL    |                |
| is_cc      | tinyint(1)       | YES  |     | NULL    |                |
| is_bcc     | tinyint(1)       | YES  |     | NULL    |                |
+------------+------------------+------+-----+---------+----------------+
```

1. How many emails did each person receive each day?

```
"""SELECT recipient, count(distinct(message_id)) AS cnt, DATE(email_date)
            FROM recipient GROUP BY recipient,DATE(email_date) ORDER BY cnt DESC;"""
```

Recipient describe all the relationship in every email.
ANSWER: Please check the text file “output_question_one.txt”

2. Let's label an email as "direct" if there is exactly one recipient and "broadcast" if it has multiple recipients. Identify the person (or people) who received the largest number of direct emails and the person (or people) who sent the largest number of broadcast emails.

DIRECT EMAILS---
We assume that if the email recipient is only one distinct person, this is a direct email.

```
"""SELECT recipient, count(distinct(message_id)) AS cnt FROM recipient WHERE message_id
             IN (SELECT message_id AS count_r FROM recipient GROUP BY message_id
             HAVING count(distinct(recipient)) = 1) GROUP BY recipient ORDER BY cnt
             DESC;"""
```

First we create a table that contains message_id which only has “ONE” distinct recipient,
And reselect recipient, message_id based on the message_id we got above and group them by recipient and count the message_id.

NOTE: count(distinct(recipient)) = 1
          count(distinct(message_id))

In the txt file there are some edge case. For example, A send email to B also cc and bcc. There will be three rows in “recipient” table and it describe the relationship either it is “to”, “cc” , and “bcc”.
That’s why I use “count(distinct(recipient)) = 1” and “count(distinct(message_id))”

BROADCAST EMAILS---
We assume that if the email recipient is more than one distinct person, this is a broadcast email.

```
"""SELECT sender, count(message_id) AS count_email
                   FROM (SELECT message_id, sender, count(distinct(recipient)) AS count_r
                   FROM recipient GROUP BY message_id, sender HAVING count_r > 1) as t1
                   GROUP BY sender ORDER BY count_email DESC;"""
```

First we create a table contains message_id ,sender that filter out only one recipient message_id, group it by sender and count the message_id by descending order, and choose the first one..

NOTE:
count(distinct(recipient))
Because we assume that there is only one distinct recipient either in “to”,”cc”, or “bcc”.
Some of the cases, some senders send an email to one receiver and the receiver’s name show up ALL in “to”,”cc” and “bcc”.

The person who received the largest number of direct emails
ANSWER: maureen.mcvicker@enron.com
The person received “115” direct emails

The person who sent the largest number of broadcast emails
ANSWER: The steven.kean@enron.com
The person sent total “253” broadcast emails

3. Find the five emails with the fastest response times. (A *response* is defined as a message from one of the recipients to the original sender whose subject line contains all of the words from the subject of the original email, and the response time should be measured as the difference between when the original email was sent and when the response was sent.)

```
"""SELECT distinct(rone.message_id), rone.subject, rone.sender, rone.recipient,
                   rtwo.message_id, rtwo.subject, rtwo.sender, rtwo.recipient,
                   TIMESTAMPDIFF(SECOND, rone.email_date , rtwo.email_date)
```

```
FROM recipient rone INNER JOIN recipient rtwo WHERE rone.sub_md5 IS NOT
NULL AND rtwo.sub_md5 IS NOT NULL
AND rone.sub_md5 = rtwo.sub_md5 AND rone.message_id != rtwo.message_id
AND TIMESTAMPDIFF(SECOND, rone.email_date , rtwo.email_date) >= 0 AND
rone.sender = rtwo.recipient
AND rtwo.sender = rone.recipient AND rone.sender != rone.recipient
ORDER BY TIMESTAMPDIFF(SECOND, rone.email_date , rtwo.email_date) ASC
limit 5;"""
```

We use the recipient table and INNER JOIN by itself. Recipient table “rone” and Recipient table “rtwo”

The filter is that we check every row in rone and rtwo, rone sender = rtwo recipient ,rone recipient = rtwo sender

1. rone sender != rone recipient (I check that there are some cases some sender send email to themselves so I need to filter them out)
2. rone hashed subject = rtwo hashed subject and both of them should not be NULL
3. The time difference between rone email time and rtwo email time >= 0 (It can filter out the negative which also mean the two row in each table switch and they are duplicate)
4. distinct(rone.message_id) Because there are some special cases that if A send to B in “to” and “cc”. There will be two data which satisfied that condition so we also need to filter it out.

ANSWER: Please check the text file “output_question_three.txt”

The text file include reply email message_id, reply subject, reply sender, reply recipient
original email message_id, original subject, original sender, original recipient
time_difference

I also list the filename

1. 228996.txt reply 228911.txt in 236 seconds
2. 121747.txt reply 121748.txt in 240 seconds
3. 122923.txt reply 122926.txt in 240 seconds
4. 228981.txt reply 228996.txt in 322 seconds
5. 221669.txt reply 199911.txt in 360 seconds

Further discussion:
It seems like I didn’t use the email table for answering the question above.
However there are some interesting points in email table;
We can make “email” table more meaningful.

```
SELECT * FROM email where message_id NOT IN (SELECT message_id FROM recipient);
SELECT count(*) FROM email WHERE label = "no recipient";
```

We found that there are “144” emails with no recipient.

```
SELECT sender,count(*) AS cnt FROM email where message_id NOT IN (SELECT
message_id FROM recipient) GROUP BY sender ORDER BY cnt DESC;
```

We found that “steven.kean@enron.com” has “130” emails with no recipient.
“legalonline-compliance@enron.com” has 5 emails with no recipient