Assignment 1

Name: Make Zhang; Student ID: 1002376555
Name: Ruiyuan Xie; Student ID: 1001705933
Course section: CSC343

Jan. 31, 2018

Schema

Relations

- User(<u>uid</u>, name, profile, email, phone, photo, lastSeen)
- Contact(<u>user</u>, <u>contact</u>, start)
- Group(<u>gid</u>, <u>uid</u>)
- Message(<u>mid</u>, from, to, content, time)
- Attachment(<u>mid</u>, <u>contentId</u>, size)
- Privacy(<u>uid</u>, lastSeen, photo, profile); (none, contacts, everyone)
- Status(<u>mid</u>, <u>uid</u>, status)

Integrity constraints

- Contact[user] $\subseteq$ User[uid]
- Contact[contact] $\subseteq$ User[uid]
- Group[uid] $\subseteq$ User[uid]
- Group[gid] $\cap$ User[uid] $= \emptyset$
- Message[from] $\subseteq$ User[uid]
- Attachment[mid] $\subseteq$ Message[mid]
- Attachment[size] ≤ 16
- Privacy[uid]] $\subseteq$ User[uid]
- Status[mid] $\subseteq$ Message[mid]
- Status[uid] $\subseteq$ User[uid]
- Status[status] $\subseteq$ { “sent”, “delivered”, “read” }
- Privacy[lastSeen] $\subseteq$ { “none”, “contacts”, “everyone” }
- Privacy[profilePhoto] $\subseteq$ { “none”, “contacts”, “everyone” }
- Privacy[profile] $\subseteq$ { “none”, “contacts”, “everyone” }

Part 1. Queries

1.1 Find all the users who have never sent a message, but who have been sent at least one message. The message may have been sent to the user or to a group that the user belongs to. Report each user id.

-Users who never sent messages

$$NeverSentMsg(uid) := \Pi_{uid} User - \rho_{Temp(uid)} \Pi_{from} Message$$

-Users who never sent message but have been sent a direct messages

$$\mathit{NeverSentButDirect}(\mathit{uid}) := \rho_{Temp2(uid)} \Pi_{to} \mathit{Message} \cap \mathit{NeverSentMeg}$$

-Users who never sent message but have been sent messages from group

$$\mathit{NeverSentButGroup}(\mathit{uid}) := \Pi_{uid}((\mathit{Group} \bowtie \mathit{NeverSentMsg}) \bowtie \rho_{Temp3(mid)} \Pi_{to} \mathit{Message})$$

-Users who have never sent a message, but who have been sent at least a message.

$$\mathit{Answer}(\mathit{uid}) := \mathit{NeverSentButDirect} \cup \mathit{NeverSentButGroup}$$

1.2 Find the users (and return their uid) who sent two or fewer messages in 2017.

-Users who sent at leat Three messages

$$AtLeastThree(from) := \Pi_{from}$$
$$(\sigma_{M1.from=M2.from=M3.from \wedge M1.mid \neq M2.mid \neq M3.mid \wedge M1.time.year=M2.time.year=M3.time.year=2017}$$
$$[(\rho_{M1}Message) \times (\rho_{M2}Message) \times (\rho_{M3}Message)])$$

-Users who sent at most two messages (Final Answer)

$$AtMostTwo(uid) := \Pi_{uid}User - \rho_{Temp(uid)}AtLeastThree$$

1.3 Find the largest group. Report the group id. If there is a tie, report them all.

This question is not solvable.

1.4 Find privacy fanatics, that is, any user who has all her privacy settings set to none and who has never sent a message to another user who has privacy settings different than her own (meaning different than all none). Note that a private user (settings are all none) who has never sent a message would be considered a privacy fanatic. Return the user's uid and name.

- All users that set all their privacy status to "none"

$$Private(uid) := \Pi_{uid}\sigma_{lastSeen="none" \wedge photo="none" \wedge profile="none"} Privacy$$

- Other users that do not set all their privacy status to "none"

$$Other(uid) := \Pi_{uid}\sigma_{lastSeen\neq"none" \vee photo\neq"none" \vee profile\neq"none"} Privacy$$

- Users with private status all "none", but they did sent Direct Message

$$\begin{aligned} PrivateButSentDirectMessage(uid) := \rho_{Temp(uid)}(\Pi_{from}(& \\ & (Message \bowtie \rho_{Temp2(to)} Other) \\ & \bowtie \rho_{Temp3(from)} Private \\ &)) \end{aligned}$$

- All group that exist at least one non-all-private users.

$$GroupWithPublic(gid) := \Pi_{gid}(Group \ - \ (Group \bowtie Private))$$

- Users with private status all "none", but they did sent Group Message to Other

$$\begin{aligned} PrivateButSentGroupMsg(uid) := \rho_{Temp(uid)}(\Pi_{from}(& \\ & (Message \bowtie \rho_{Temp2(to)} GroupWithPublic) \\ & \bowtie \rho_{Temp3(from)} Private \\ &)) \end{aligned}$$

- All users that are privacy fanatics

$$\begin{aligned} Answer(uid, name) := \Pi_{uid,name}([& Private(uid) - \\ & PrivateButSentDirectMessage(uid) - \\ & PrivateButSentGroupMsg(uid)] \ \bowtie User) \end{aligned}$$

1.5 Consider only users whose privacy settings state that everyone may see their lastSeen time (lastSeen = "everyone"). Among such users, report the uid, name and lastSeen of the user(s) whose lastSeen time is the most recent. Since times are ordered, the most recent time is the largest time value. If there are ties, report all users. These users have the most recent public lastSeen time.

- Users who have lastSeen setting to everyone.

$$Public(uid, lastSeen, name) := \Pi_{uid,\ lastSeen,\ name}(\Pi_{uid}(\sigma_{privacy=\text{"}everyone\text{"}}(Privacy)) \bowtie User)$$

- Users who doesn't have the most recent lastSeen time.

$$Losers(uid, lastseen, name) := \Pi_{P1.uid,\ P1.lastSeen,\ P1.name}(\Pi_{uid}(\sigma_{P1.lastSeen<p2.lastSeen}[(\rho_{P1}Public) \times (\rho_{P2}Public)])$$

- Users who did have the most recent lastSeen time(Final answer)

$$Winners(uid, lastSeen, name) := \Pi_{uid,\ lastSeen,\ name}User\ -\ Losers$$

1.6 A user's contact list can be sorted by the start time. Find users who send their first direct message to a contact in the same order as the contact list order. So if Sue is Pat's oldest contact and Jo is the second oldest contact, then Pat's first direct message to Sue happens before her first direct message to Jo and so on for all contacts. Include users with empty contact lists. Return user's uid.

- Get all direct Messages

$$DirectMsg(to, from, time, mid) := \Pi_{to,\ from,\ time,\ mid}(Message \bowtie (\rho_{Temp(to)}\Pi_{uid}User))$$

- Get mids that are not the first messages

$$Fail(mid) := \Pi_{d1.mid}(\sigma_{d1.to=d2.to \wedge d1.from=d2.from \wedge d1.time>d2.time}[(\rho_{d1}DirectMsg) \times (\rho_{d2}DirectMsg)])$$

- Get first messages

$$FirstMsg(from, to, time) := \Pi_{from,\ to,\ time}((\Pi_{mid}DirectMsg\ -\ Fail) \bowtie DirectMsg)$$

- Combine the first messages with contact information

$$FirstMsgWithContact(user, contact, time, start) := (\rho_{temp(user,\ contact,\ time)}FirstMsg) \rhd\lhd \Pi_{user,\ contact,\ start}Contact$$

- Find all users doesn't follow the time restirction

$$\begin{aligned} Loser(uid) := \rho_{temp(uid)}(& \\ & \Pi_{user} \\ & \sigma_{F1.user=F2.user\ \wedge\ F1.contact\neq F2.contact\ \wedge\ F1.start<F2.start\ \wedge\ F1.time>F2,time} \\ & [(\rho_{F1}FirstMsgWithContact) \times (\rho_{F2}FirstMsgWithContact)] \\ &) \end{aligned}$$

- The rest users will be the answer

$$Ans(uid) := \Pi_{uid}User\ -\ Loser(uid)$$

1.7 Return all pairs of users with the same name. Return the two uids and the common name. Return each pair only once. (For example, if user 1 and user 2 are both named 'Pat', then return either [1, 2, 'Pat'] or [2, 1, 'Pat'] but not both).

- Get the users who have the same name but with different uid

$$\textit{Answer(U1.uid, U2.uid, name)} := \Pi_{U1.uid,\ U2.uid,\ U1.name}$$
$$\sigma_{U1.uid > U2.uid\ \wedge\ U1.name = U2.name}$$
$$[(\rho_{U1} User) \times (\rho_{U2} User)]$$

1.8 For each user and contact, report the time that the first direct message was sent from the user to the contact and the time the last direct message was sent. Return the uid of the user (in an attribute named user) and the contact (in an attribute named contact) and the first time (earliest) (in an attribute named first) and last (most recent) time (in an attribute named last). If a user has not sent any direct messages to a contact then include the user and contact with the value 0 for both the first and last times.

- Get the messages that are not last

$$NotLast(from, to, time) := \Pi_{M1.from,\ M1.to,\ M1.time} \sigma_{M1.from=M2.from\ \wedge\ M1.to=M2.to\ \wedge\ M1.time<M2.time,\ M1.time} [(\rho_{M1}Message) \times (\rho_{M2}Message)]$$

- The rest will be Last time messages

$$LastTime(from, to, last) := \rho_{temp(from,\ to,\ last)}(\Pi_{from,\ to,\ time}Message\ -\ NotLast)$$

- Get the messages that are not first

$$NotFirst(from, to, time) := \Pi_{M1.from,\ M1.to,\ M1.time} \sigma_{M1.from=M2.from\ \wedge\ M1.to=M2.to\ \wedge\ M1.time>M2.time,\ M1.time} [(\rho_{M1}Message) \times (\rho_{M2}Message)]$$

- The rest will be First time messages

$$FirstTime(from, to, first) := \rho_{temp(from,\, to,\, first)}(\Pi_{from,\, to,\, time}Message - NotFirst)$$

- Find the users that are contacts but never send messages and set their first and last time as 0.

$$NoContact(from, to, first, last) := \Pi_{from,\, to,\, 0,\, 0}(\rho_{temp(from,to)}\Pi_{user,\, contact}Contact - \Pi_{from,\, to}Message)$$

- Combine those information to get the final answer

$$Answer(uid, contact, first, last) := \rho_{temp(uid,\, contact,\, from,\, last)}((FirstTime \bowtie LastTime) \cup NoContact)$$

1.9 A 'spammer' is a user who posts unwanted direct messages that are not read. A spammer must have sent at least direct message (so this message will appear in the Status relation). Because users may not be aware that someone is a spammer, they may read some of their initial messages. However, once they decide a certain user is a spammer, the receivers stop reading all messages from the spammer. This means that for a user who is sent a direct message from a spammer there are no delivered messages with a time that is earlier than any read message from the spammer. Return the spammer's user id and all their privacy settings (Privacy.lastSeen, Privacy.photo, Privacy.profile). Do not consider groups for this question. Only consider direct messages sent from a user to another single user (not to a group).

- Get all messages with their sending status

$$MsgWithStatus(mid, from, to, time, status) := \Pi_{mid,\ from,\ to,\ time,\ status}[Message \bowtie \rho_{temp(mid,\ to,\ status)}Status]$$

- Find messages that haves more than one status

$$MoreThanOne(from,\ to,\ time,\ status) := \Pi_{M1.from,\ M1.to,\ M1.time,\ M1.status}\ \sigma_{M1.from=M2.from\ \wedge\ M1.to=M2.to\ \wedge\ (M1.status\neq"read"\ \wedge\ M2.status="read")\ \vee\ (M1.status="read"\ \wedge\ M2.status\neq"read")}[\rho_{M1}MsgWithStatus \times \rho_{M2}MsgWithStatus]$$

- Get messages that are either all read or unread formed by the same sender

$$OnlyOneEitherAllReadOrUnread(from,\ to,\ time,\ status) := \Pi_{from,\ to,\ time,\ status}MsgWithStatus\ -\ MoreThanOne$$

- if messages that have sent by the same sender are all readed, the sender must be safe(not spammer)

$$OnlyOneAllRead(from) := \Pi_{from}\sigma_{status="read"}OnlyOneEitherAllReadOrUnread$$

- if messages that have sent by the same sender are all unreaded, the sender must be unsafe(spammer)

$$OnlyOneAllNotRead(from) := \Pi_{from}\sigma_{status\neq"read"}OnlyOneEitherAllReadOrUnread$$

- For those people who send messages with more than one status, if there is a recevier who read all his messages, the sender must not be a spammer.

$$HaveOnePersonAllRead(from) := \Pi_{from}MoreThanOne - \Pi_{m1.from}\ \sigma_{m1.from=m2.from\ \wedge\ m1.to=m2.to\ \wedge\ (m1.status\neq"read"\ \vee\ s2.status\neq"read")}[\rho_{m1}(MoreThanOne) \times \rho_{m2}(MoreThanOne)]$$

- Find the sender who send messages with more than one status, and there is no person who read all his messages. They may become potential spammer.

$$MoreThanOneNotAllRead(from,\ to,\ time,\ status) := MoreThanOne - (MoreThanOne \rtimes\!\!\ltimes HaveOnePersonAllRead)$$

-Find the messages that **exist a read after unread**

$$ExistReadAfterUnread(from) := \Pi_{m1.from}\ \sigma_{m1.from=m2.from\ \wedge\ m1.to=m2.to\ \wedge\ m1.status="read"\wedge m2.status\neq"read"\wedge m1.time>m2.time}[\rho_{m1}(MoreThanOneNotAllRead) \times \rho_{m2}(MoreThanOneNotAllRead)]$$

- Find spammer who send all unread messages after at least one read.

$$NoReadAfterUnread(from) := \Pi_{from}(MoreThanOneNotAllRead) - ExistReadAfterUnread$$

- The final answer will be all people except safe people and intersect with all spammers.

$$\begin{aligned}&Answer(uid, lastSeen, photo, profile) :=\\ &\qquad (\rho_{answer(uid)}[(\Pi_{from}MsgWithStatus \ - \ OnlyOneAllRead \ - \ HaveOnePersonAllRead)\\ &\qquad \cap\\ &\qquad (OnlyOneAllNotRead \cup NoReadAfterUnread)]) \bowtie Privacy\end{aligned}$$

Part 2. Integrity Constraints

2.1 The receiver (Message[to]) of a message must be either a user (User[uid]) or a group (Group[gid]).

- Messages won't contains other ids.

$$\Pi_{to} Message \; - \; \rho_{temp1(to)} \Pi_{uid} User \; - \; \rho_{temp2(to)} \Pi_{gid} Group = \varnothing$$

2.2 A user can only send messages to users in her contacts (Contact[contact]) and the time of the message must be after the start of the contact. This includes direct messages sent to a user and messages sent to a group. All members of the group must be in the user's contacts.

- Direct Messages that violate the start and send time restriction

$$WrongTimeDirectMsg(from) := \Pi_{from} \sigma_{Message.from=Contact.user \; \wedge \; Message.to=Contact.contact \; \wedge \; Message.time<Contact.start} [Message \times Contact]$$

- Combine all group members' uid with the Messages realation

$$GroupMsgWithMembers(from,\; uid,\; time) := \Pi_{from,uid,time}[(\rho_{temp(to,\; uid)} Group) \times Message]$$

-Group Messages that violate the start and send time restriction

$$WrongTimeGroupMsg(from) := \Pi_{from} \sigma_{M.from=Contact.user \; \wedge \; M.to=Contact.contact \; \wedge \; M.time=Contact.start} [(\rho_M GroupMsgWithMembers) \times Contact]$$

-Messages that have been sent but the sender and receiver are not contact, which violate the requirement

$$MsgBetweenNoContact(from) := \Pi_{from} (\Pi_{from,\; uid}[Status \bowtie Message] \; - \; \rho_{temp(from,\; uid)} \Pi_{user,\; contact} Contact)$$

-All those three wrong situations should not exist

$$WrongTimeDirectMsg \cup WrongTimeGroupMsg \cup MsgBetweenNoContact = \varnothing$$

2.3 The total size of all attachments in a message must be less than 128MB.

This question is not solvable.

2.4 The Status relation may not contain a message and user if the message was not sent to that user (either directly or the user was part of a group that received the message).

- After minus all the possible messages relations(either direct or group messages), Status will not contains other informations

$$\Pi_{mid,\; uid}(Status) - \Pi_{mid,\; uid}[(\rho_{temp(mid,\; uid)} \Pi_{mid,\; to} Message) \bowtie (User)] - \Pi_{mid,\; uid}[(Message) \bowtie (\rho_{temp(to,\; uid)} Group)] = \varnothing$$