CSC343 Assignment 1

Quan Xu, Zhiyang Yu, Zijin Zhang

Due: Mon Feb 4th, 11:59 p.m.

Part 1: Queries

1

Pair up all the users
$Pair(u1, u2) := \Pi_{u1.uid, u2.uid} \sigma_{u1.uid > u2.uid} [(\rho_{u1} Users) \times (\rho_{u2} Users)]$

Find the user pairs that u1 does not follow u2
$NotFollowed(u1, u2) := Pair - \rho(u1, u2)(\Pi_{follower, followed} Follows)$

Find the user who liked the post of unfollowed user
$LikedUnfollowedPost(user) := \Pi_{u1} \sigma_{u2=uid} NotFollowed \times (Likes \bowtie Post)$

Find the user who viewed the story of unfollowed user
$ViewedUnfollowedStory(user) :=$

$\Pi_{u1} \sigma_{(NotFollowed.u1=Saw.viewerid) \wedge (NotFollowed.u2=Story.uid) \wedge (Saw.sid=Story.sid)} (Saw \times Story \times NotFollowed)$

Find the user who did at least one of the things above
$DoAny(uid) := LikedUnfollowedPost \cup ViewedUnfollowedStory$

User who did not do one of the things list above
$ResultUser(uid) := \Pi_{uid} Users - DoAny$

Gather information and rename to get the final answer
$Answer(name, description) := \Pi_{name, about}(ResultUser \bowtie Users)$

2

Find all Hashtag in 2018 with its $pid, tagandwhen$
$Tags2018(pid, tag, when) := \Pi_{pid, tag, when} \sigma_{when.year=''2018''} (Post \bowtie Hashtag)$

Find all Hashtags in 2018 that appear at least twice on a day of 2018
$AppearedTwice(tag, when) :=$
$\Pi_{T1.tag, T1.when} \sigma_{(T1.tag=T2.tag) \wedge (T1.when.day=T2.when.day) \wedge (T1.pid>T2.pid)} [(\rho_{T1} Tags2018) \times (\rho_{T2} Tags2018)]$

3

Find all reciprocal followers:
$ReciprocalFollowers(u1, u2) := \Pi_{F1.followed, F2.followed}$

$\sigma_{(F1.follower=F2.followed) \wedge (F2.follower=F1.followed) \wedge (F1.followed<F2.followed)} [(\rho_{F1} Follows) \times (\rho_{F2} Follows)]$

Find all the users followed $u1$ and $u2$ respectively.
$FollowedU1(u1, u2, follower) :=$

$\Pi_{RF.u1, RF.u2, F.follower} \sigma_{RF.u1=F.followed} [(\rho_{RF} ReciprocalFollowers) \times (\rho_F Follows)]$

$FollowedU2(u1, u2, follower) :=$

$$\Pi_{RF.u1,RF.u2,F.follower}\sigma_{RF.u2=F.followed}[(\rho_{RF}ReciprocalFollowers) \times (\rho_F Follows)]$$

$FollowedU1orU2 := (FollowedU1 \cup FollowedU2) - (FollowedU1 \cap FollowedU2)$

$Answer(u1, u2, name, email) :=$
$$\Pi_{F.u1,F.u2,User.name,User.email}\sigma_{F.follower=User.uid}[(\rho_F FollowedU1orU2) \times User]$$

4

Cannot be expressed.

5

Find pair of Reciprocal:
$ReciprocalFollowers(follower_1, follower_2) :=$
$\Pi_{F_1.follower,F_2.follower}\sigma_{F1.follower<F_2.follower \wedge F1.follower=F_2.followed \wedge F_2.follower=F_1.followed}(\rho_{F_1}Follows \times \rho_{F_2}Follows)$

Find $follower_1$'s post:
$Follower_1Post(pid, follower_1, follower_2) := \Pi_{pid,follower_1,follower_2}\sigma_{uid=follower_1}(ReciprocalFollowers \times Post)$

Find $follower_1$'s post which $follow_2$ likes:
$Follower_2Like(pid, follower_1, follower_2) :=$
$\Pi_{pid,follower_1.follower_2}\sigma_{uid=follower_1 \wedge liker=follower_2 \wedge Post.pid=Likes.pid}(ReciprocalFollowers \times Post \times Likes)$

Find $follower_1$'s post which $follower_2$ does not like:
$Follower_2NotLike(pid, follower_1, follower_2) := Follower_1Post - Follower_2Like$

Find $follower_2$'s post:
$Follower_2Post(pid, follower_1, follower_2) := \Pi_{pid,follower_1,follower_2}\sigma_{uid=follower_2}(ReciprocalFollowers \times Post)$

Find $follower_2$'s post which $follow_1$ likes:
$Follower_1Like(pid, follower_1, follower_2) :=$
$\Pi_{pid,follower_1.follower_2}\sigma_{uid=follower_2 \wedge liker=follower_1 \wedge Post.pid=Likes.pid}(ReciprocalFollowers \times Post \times Likes)$

Find $follower_2$'s post which $follower_1$ does not like:
$Follower_1NotLike(pid, follower_1, follower_2) := Follower_2Post - Follower_1Like$

Find back backscratchers:
$Backscratchers(follower_1, follower_2) :=$
$ReciprocalFollowers - (\Pi_{follower_1,follower_2}Follower_2NotLike) - (\Pi_{follower_1,follower_2}Follower_1NotLike)$

Find user who follow some pair of backscratcher users:
$Answer(follower) :=$
$$\Pi_{F_1.follower}\sigma_{F_1.follower=F_2.follower \wedge F_1.followed=Backscratchers.follower_1 \wedge F_2.followed=Backscratchers.follower_2}$$
$$(\rho_{F_1}Follows \times \rho_{F_2}Follows \times Backscratchers)$$

6

Find activities with uid and when(time):
$Activity(uid, when) := (\Pi_{uid,when}Post) \cup (\Pi_{uid,when}Story)$

Find activities that not is the latest:
$NotLatest(uid, when) := \Pi_{A1.uid,A1.when}\sigma_{(A1.uid=A2.uid) \wedge (A1.when<A2.when)}[(\rho_{A1}Activity) \times (\rho_{A2}Activity)]$

Find activities that is the latest:
$Latest(uid, when) := Activity - NotLatest$

Find the followed and their latest activity date of each user:
$LatestAndFollows(follower, followed, when) := \Pi_{follower,uid,when}\sigma_{Follows.followed=Latest.uid}[Latest \times Follows]$

Find not most recent followed for each follower:
$NotMostRecent(follower, followed, when) :=$
$\Pi_{L1.follower,L1.followed,L1.when}\sigma_{(L1.follower=L2.follower)\wedge(L1.when<L2.when)}[(\rho_{L1}LatestAndFollows)\times(\rho_{L2}LatestAndFollows)]$

Find most recent followed for each follower:
$MostRecent(follower, followed, when) := LatestAndFollows - NotMostRecent$

$Answer(follower, followed, email, when) :=$
$\Pi_{F1.name,F2.name,F2.email,when}\sigma_{(F1.uid=MostRecent.follower)\wedge(F2.uid=MostRecent.followed)}[(\rho_{F1}User)\times(\rho_{F2}User)\times MostRecent]$

7

Cannot be expressed.

8

For each user who has ever put any comments, report their id and the id of the first and of the last post they commented on.

Comment that not last:
$NotLast(commenter, pid) := \Pi_{c_1.commenter,c_1.pid}\sigma_{c_1.commenter=c_2.commenter\wedge c_1.when<c_2.when}(\rho_{c_1}Comment \times \rho_{c_2}Comment)$

Comment that last:
$Last(commenter, last\_pid) := (\Pi_{Commenter,Pid}Commenter) - NotLast$

Comment that not first:
$NotFirst(commenter, pid) := \Pi_{c_2.commenter,c_2.pid}\sigma_{c_1.commenter=c_2.commenter\wedge c_1.when<c_2.when}(\rho_{c_1}Comment \times \rho_{c_2}Comment)$

Comment that first:
$First(comment, first\_pid) := (\Pi_{commenter,pid}Commenter) - NotLast$

Combine together:
$Answer(Commenter, first\_pid, last\_pid) := First \bowtie Last$

Part 2.

1. A view on a story must occur after the date-time of the story itself. (Remember that you can compare two date-time attributes with simple <, >= etc.)

ANSWER: The set of story where time of one of its view is before time of the story is empty.
$\sigma_{vTime<=sTime}(\rho_{sid,tmp_1,sTime,tmp_2}Story) \bowtie (\rho_{tmp_3,sid,vTime}Saw) = \emptyset$

2. Each user can have at most one current story.

ANSWER: The set of two stories' that occur at same time with same use (same uid) is empty.
$\sigma_{s_1.uid=s_2.uid\wedge s_1.sid<s_2.sid\wedge s_1.current=s_2.current=True}(\rho_{s_1}Story \times \rho_{s_2}Story) = \emptyset$