

# CSC343 Assignment1

Haoda Li, Xinyi Liu, Kewei Qiu

February 4, 2019

## Part I

1. *# Find all (uid1, uid2) tuples such that user1 ever liked user2's posts .*  
LikeUidsTuple(uid1, uid2) :=  $\rho_{(uid1, uid2)}(\Pi_{liker, uid} (Likes \bowtie Post))$   
*# Find all uid1 such that user1 liked user2's post and also follows user2*  
LikeAndFollowUids(uid1) :=  $\Pi_{uid1} (LikeUidsTuple \cap \rho_{(uid1, uid2)}(\Pi_{follower, followed} (Follows)))$   
*# Find all uid1 such that user1 liked user2's post but doesn't follow user2*  
LikeAndUnfollowUids(uid1) :=  $\Pi_{uid1} (LikeUidsTuple) - LikeAndFollowUids$   
*# Find all uid such that user1 have never liked a post of a user that they do not follow*  
NeverLikeFollowingUids(uid1) :=  $\Pi_{uid} (User) - LikeAndUnfollowUids$   
*# Find all (uid1, uid2) tuples such that that user1 ever viewed user2's stories .*  
ViewUidsTuple(uid1, uid2) :=  $\rho_{(uid1, uid2)}(\Pi_{viewerid, uid} (Saw \bowtie Story))$   
*# Find all uid1 such that user1 viewed user2's stories and also follows user2*  
ViewAndFollowUids(uid) :=  $\Pi_{uid1} (ViewUidsTuple \cap \rho_{(uid1, uid2)}(\Pi_{follower, followed} (Follows)))$   
*# Find all uid1 such that user1 viewed user2's story but doesn't follow user2*  
ViewAndUnfollowUids(uid1) :=  $\Pi_{uid1} (ViewUidsTuple) - ViewAndFollowUids$   
*# Find all uid such that user1 have never viewed a story of a user that they do not follow*  
NeverViewFollowingUids(uid1) :=  $(\Pi_{uid} (User)) - ViewAndUnfollowUids$   
*# Find all uid such that user1 have never liked or viewed a post or a story of a user that they do not follow*  
AnswerUids(uid) :=  $NeverLikeFollowingUids \cup NeverViewFollowingUids$   
*# Report the selected uid's name and about as username and description*  
Answer(username, description) :=  $\rho_{(username, description)}(\Pi_{name, about} (User \bowtie AnswerUids))$
2. *# Match posts and hashtag they mentioned in captions in 2018*  
PostIn18(pid, when, tag) :=  $\Pi_{pid, when, tag} ((\sigma_{when.year = 2018} (Post)) \bowtie (Hashtag))$   
*# Find all pairs of posts which have the same hashtag in the same day in 2018*  
PostPairsIn18(P1.pid, P2.pid, when, tag) :=  $\sigma_{(P1.pid < P2.pid) \wedge (P1.tag = P2.tag) \wedge (P1.when.date = P2.when.date)}$   
 $(\rho_{P1} (PostIn18) \times \rho_{P2} (PostIn18))$   
*# Pick out tags*  
Result(tag) :=  $\Pi_{tag} (PostPairsIn18)$
3. *# Find all "reciprocal followers"*  
Reciprocal(r1, r2) :=  $\Pi_{uid1, uid2} ((\rho_{F1}(uid1, uid2)(Follows)) \bowtie_{uid1=uid4 \wedge uid2=uid3} (\rho_{F2}(uid3, uid4)(Follows)))$   
*# Find all uncommon follower's uids*  
Uncommon(uid, r1, r2) :=  $\Pi_{follower, r1, r2}$   
 $(Follows \bowtie_{follower \neq r1 \wedge follower \neq r2 \wedge ((followed = r1 \wedge followed \neq r2) \vee (followed \neq r1 \wedge followed = r2))} Reciprocal)$   
*# Report the uncommon follower's uid, name, email, and the pair of reciprocal follower's uids*  
Result(uncommon\_uid, name, email, r1\_uid, r2\_uid) :=  $\Pi_{uid, name, email, r1, r2} (Uncommon \bowtie User)$
4. Cannot be expressed

5. *# Find all the pairs of users that user1 does not like all of user2's post*  

$$\text{Unlikes}(\text{uid1}, \text{uid2}) := \Pi_{\text{liker}, \text{uid}} (\text{Post} \times \rho_{\text{L}(\text{liker})} (\Pi_{\text{uid}}(\text{User})) - \text{Post} \bowtie \Pi_{\text{uid}, \text{liker}}(\text{Likes}))$$
*# Find all the pairs of users that user1 follows user2 and likes all users'2 posts*  

$$\text{FL}(\text{uid1}, \text{uid2}) := \rho_{\text{uid1}, \text{uid2}}(\Pi_{\text{follower}, \text{followed}}(\text{Follows})) - \text{Unlikes}$$
*# Find all pairs of "backscratchers"*  

$$\text{Back}(\text{b1}, \text{b2}) := \Pi_{\text{uid1}, \text{uid2}}(\rho_{\text{FL1}}(\text{uid1}, \text{uid2})(\text{FL}) \bowtie_{\text{uid1} = \text{uid4} \wedge \text{uid2} = \text{uid3}} \rho_{\text{FL2}}(\text{uid3}, \text{uid4})(\text{FL}))$$
*# Report the user id of all users who follow some pair of backscratcher users*  

$$\text{Result}(\text{uid}) := \Pi_{\text{F1. uid1}}(\sigma_{\text{F1. uid1} = \text{F2. uid1} \wedge \text{F1. uid2} = \text{b1} \wedge \text{F2. uid2} = \text{b2}}(\rho_{\text{F1}}(\text{Follows}) \times \text{Back} \times \rho_{\text{F2}}(\text{Follows})))$$
6. *# Get all users' most recent activity*  

$$\text{Action}(\text{uid}, \text{when}) := \Pi_{\text{uid}, \text{when}} (\text{Post}) \cup \Pi_{\text{uid}, \text{when}} (\text{Story})$$

$$\text{NotMostRecentAction}(\text{uid}, \text{when}) := \Pi_{\text{A1. uid}, \text{A1. when}} (\sigma_{(\text{A1. uid} = \text{A2. uid}) \wedge (\text{A1. when} < \text{A2. when})} (\rho_{\text{A1}} (\text{Action}) \times \rho_{\text{A2}} (\text{Action})))$$

$$\text{MostRecentAction}(\text{uid}, \text{when}) := \text{Action} - \text{NotMostRecentAction}$$
  
*# Get uids of each users' most recent activity user*  

$$\text{FollowedAction}(\text{follower}, \text{followed}, \text{when}) := \Pi_{\text{follower}, \text{followed}, \text{when}} (\text{Follows} \bowtie_{\text{followed} = \text{uid}} \text{MostRecentAction})$$

$$\text{NotMostRecent}(\text{follower}, \text{followed}, \text{when}) := \Pi_{\text{F1. follower}, \text{F1. followed}, \text{F1. when}} (\sigma_{(\text{F1. follower} = \text{F2. follower}) \wedge (\text{F1. when} < \text{F2. when})} (\rho_{\text{F1}} (\text{FollowedAction}) \times \rho_{\text{F2}} (\text{FollowedAction})))$$

$$\text{MostRecent}(\text{follower}, \text{followed}, \text{when}) := \text{FollowedAction} - \text{NotMostRecent}$$
  
*# Report the required information*  

$$\text{FollowerInfo}(\text{follower}, \text{followerName}) := \Pi_{\text{uid}, \text{name}} (\text{User})$$

$$\text{FollowedInfo}(\text{followed}, \text{followedName}, \text{email}) := \Pi_{\text{uid}, \text{name}, \text{email}} (\text{User})$$

$$\text{Result}(\text{followerName}, \text{followedName}, \text{email}, \text{when}) := \Pi_{\text{followerName}, \text{followedName}, \text{email}, \text{when}} (\text{MostRecent} \bowtie \text{FollowerInfo} \bowtie \text{FollowedInfo})$$
7. Cannot be expressed
8. *# Remove attribute text from relation Comment*  

$$\text{NoText}(\text{pid}, \text{commenter}, \text{when}) := \Pi_{\text{pid}, \text{commenter}, \text{when}} (\text{Comment})$$
*# Create a relation about each pairs of comments of each commenter*  

$$\text{CommentPairs}(\text{N1. pid}, \text{N2. pid}, \text{commenter}, \text{N1. when}, \text{N2. when}) := \sigma_{\text{N1. pid} < \text{N2. pid}} (\rho_{\text{N1}}(\text{N1. pid}, \text{commenter}, \text{N1. when}) (\text{NoText}) \bowtie \rho_{\text{N2}}(\text{N2. pid}, \text{commenter}, \text{N2. when}) (\text{NoText}))$$
*# Choose comments which are not first posted by each commenter*  

$$\text{NotFirst}(\text{pid}, \text{commenter}) := \Pi_{\text{N1. pid}, \text{commenter}} (\sigma_{\text{N1. when} > \text{N2. when}} (\text{CommentPairs}))$$
*# Choose comments which are not last posted by each commenter*  

$$\text{NotLast}(\text{pid}, \text{commenter}) := \Pi_{\text{N1. pid}, \text{commenter}} (\sigma_{\text{N1. when} < \text{N2. when}} (\text{CommentPairs}))$$
*# Remove "not first posted list" from comments list to get first post for each commenter*  

$$\text{First}(\text{first}, \text{commenter}) := (\Pi_{\text{pid}, \text{commenter}} (\text{Comments})) - \text{NotFirst}$$
*# Remove "not last posted list" from comments list to get last post for each commenter*  

$$\text{Last}(\text{last}, \text{commenter}) := (\Pi_{\text{pid}, \text{commenter}} (\text{Comments})) - \text{NotLast}$$
*# Use natural join to get relation about each commenters and their first and last post*  

$$\text{Result}(\text{first}, \text{last}, \text{commenter}) := \text{First} \bowtie \text{Last}$$

## Part II

1.  $\sigma_{(\text{S. sid} = \text{V. sid}) \wedge (\text{S. when} > \text{V. when})}(\rho_{\text{S}}(\text{Story}) \times \rho_{\text{V}}(\text{Saw})) = \emptyset$
2.  $\sigma_{(\text{S1. sid} < \text{S2. sid}) \wedge (\text{S1. uid} = \text{S2. uid}) \wedge (\text{S1. when} = \text{S2. when}) \wedge (\text{S1. current} = \text{S2. current} = \text{True})}((\rho_{\text{S1}}(\text{Story})) \times (\rho_{\text{S2}}(\text{Story}))) = \emptyset$