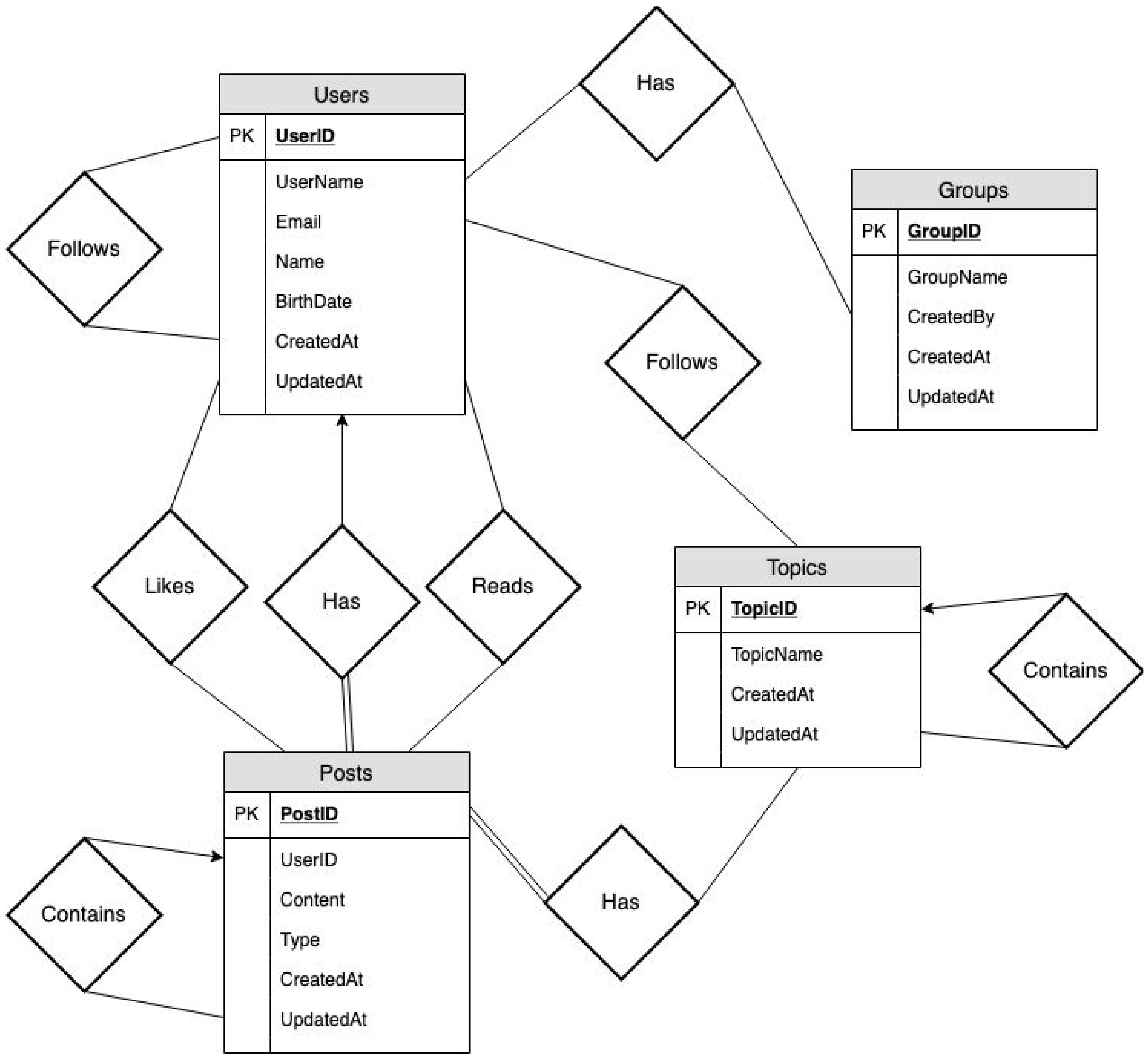
1. Create an Entity-Relationship Model of the problem space
In the case of the first, a PDF with the ER model in diagram plus description as necessary (should not be more than a couple of pages).



We have four entities in our social network including **Users**, **Posts**, **Topics** and **Groups**.

Users
userID
userName
email
name
birthDate
createdAt
updatedAt

User entity contains one primary key (userID) and six other attributes (userName, email, name, birthDate, createdAt, UpdatedAt). UserID is a unique number in the scope of the social network. Since userID can uniquely identify each user, we choose userID as the primary key of the entity. The remaining six other attributes are some additional information of user. userName is the name of the user, email is the email address of the user, birthDate is the date of birth of the user in the format of year-month-date, createdAt is the datetime of the creation of the record, updatedAt is the datetime of the update of the record. userName and email are unique.

Posts
postID
userID
content
type
createdAt
updatedAt

Post entity contains one primary key (postID) and five other attributes (userID, content, type, createdAt, updatedAt). postID is a unique number in the scope of the social network. Since postID can uniquely identify each post, we choose postID as the primary key of the post entity. The remaining five other attributes are some additional information of the post. userID is the userID of the author of the post, content is the content of the post, type is the type of the post which can be one of link, text or image, createdAt is the datetime of the creation of the record, updatedAt is the datetime of the update of the record.

Topics
topicID
topicName
createdAt
updatedAt

Topics entity contains one primary key (topicID) and three other attributes (topicName, createdAt, updatedAt). topicID is a unique number in the scope of the social network. Since topicID is able to uniquely identify a topic, we choose topicID as the primary key of the entity. The remaining three other attributes are some additional information of the topic. topicName is the name of the topic, createdAt is the datetime of the creation of the record, updatedAt is the datetime of the update of the record.

Groups
groupID
groupName
createdBy
createdAt
updatedAt

Group entity contains one primary key (groupID) and four other attributes (groupName, createdBy, createdAt, updatedAt). groupID is a unique number that can uniquely identify a group in the scope of the social network, therefore we choose groupName as the primary key of the entity Groups. The remaining four other attributes are some additional information of the group. gourpName is the name of the group, createdBy is the userID that created the group, createdAt is the datetime of the creation of the record, updatedAt is the datetime of the update of the record.

We have **nine** relationships in our social network. They are listed as follows:

1. UserFollowingUsers
userID
followerUserID
createdAt
updatedAt

UserFollowingUsers relationship is a many to many relationship, since one user can follow many users and one user can be followed by multiple users as well. userID is a foreign key from user entity and follwerUserID is a foreign key from user entity.

2. UserHasPosts
UserHasPosts relationship is a one to many relationship, since one user can create many posts but one post must be created by exactly one user. So it is a participation constraint. It is done by having userID as a foreign key referencing userID of user entity in post entity.

3. PostHasResponses
responsePostID
postID
createdAt

updatedAt

PostHasResponses relationship is a one to many relationship since one post can have multiple sub posts (responses) but one sub post (response) can have one parent post at most. postID is a foreign key from post entity and responsePostID is a foreign key from post entity as well.

4. TopicContainsSubtopics
subtopicID
topicID
createdAt
updatedAt

TopicContainsSubtopics relationship is a one to many relationship since one topic can have multiple sub topics but one sub topic can have one parent topic at most. topicID is a foreign key from topic entity and subtopicID is a foreign key from topic entity as well.

5. TopicsPosts
topicID
postID
createdAt
updatedAt

TopicsPosts relationship is a many to many relationship since one topic can have multiple posts and one post can belong to multiple topics. topicID is a foreign key from topic entity and postID is a foreign key from post entity. A post must have at least one topic. So it is a participation constraint.

6. UsersLikedPosts
userID
postID
createdAt
updatedAt

UsersLikedPosts is a many to many relationship since one user can like many posts and one post can be liked by many users. userID is a foreign key from user entity and postID is a foreign key from post entity.

7. GroupsUsers
groupID

<u>userID</u>
createdAt
updatedAt

GroupsUsers is a many to many relationship since one user can have multiple groups and one group can have multiple users. groupID is a foreign key from group entity and userID is a foreign key from user entity.

8. UsersTopics
<u>userID</u>
<u>topicID</u>
createdAt
updatedAt

UsersTopics is a many to many relationship since one user can follow many topics and one topic can be followed by multiple users. userID is a foreign key from user entity and topicID is a foreign key from topic entity.

9. UsersReadPosts
<u>userID</u>
<u>postID</u>
createdAt
updatedAt

UsersReadPosts is a many to many relationship since one user can read multiple posts and one post can be read by multiple users. userID is a foreign key from user entity and postID is a foreign key from post entity.