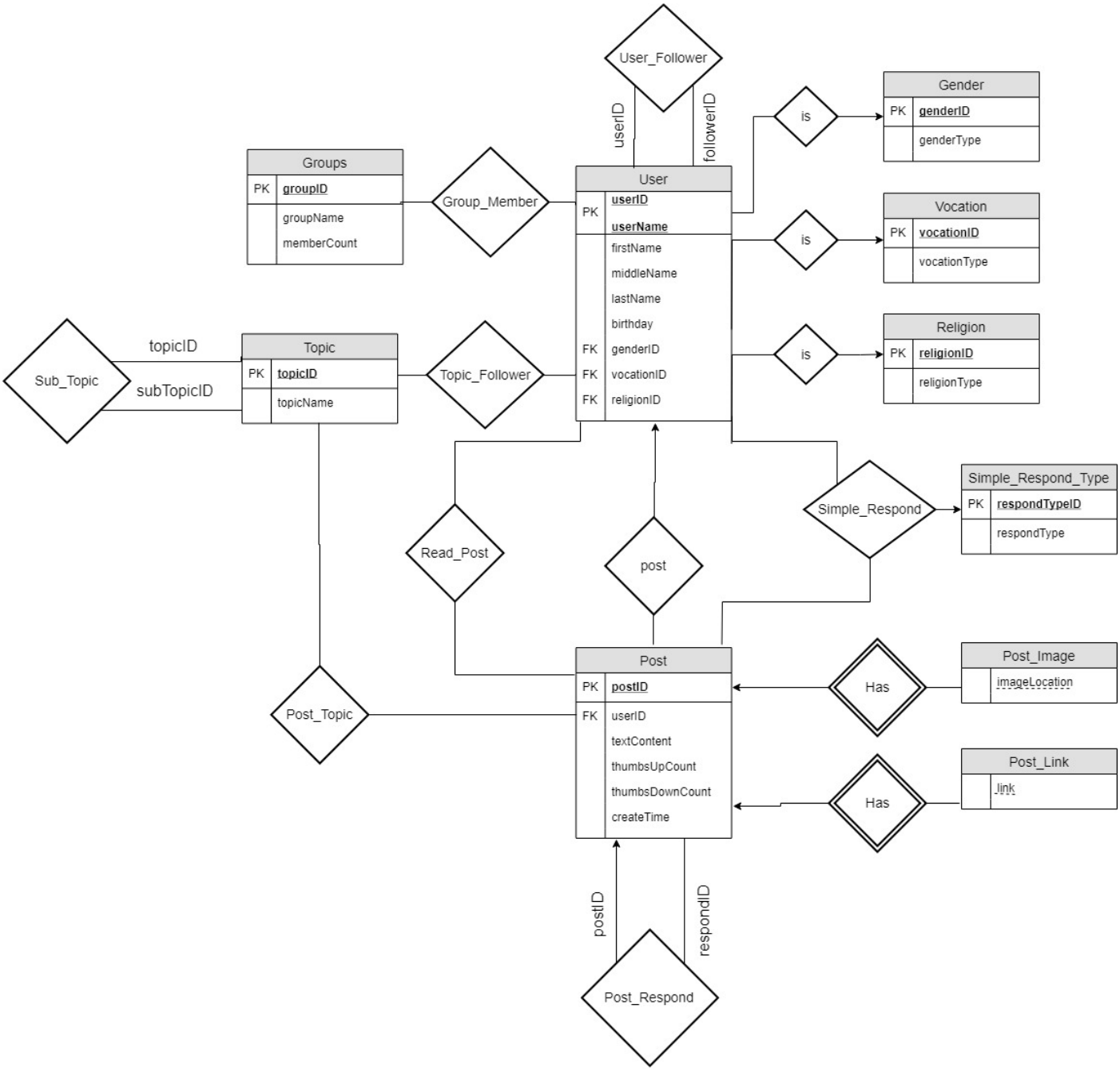
Entity-Relationship Model of Social Network

1. ER Model Diagram



2. ER Model Description

The diagram above is the ER model diagram of Social Network.

- Rectangles represent entity set.
- Diamonds represent relationship set.
- Attributes are listed inside entity rectangle.
- Underlined attributes are part of the primary key.
- Double diamonds represent the relationship of a weak entity, and underline the discriminator of a weak entity set with a dashed line.
- Directed lines (→) between entity set and relationship set represent signifying “one”.
- Undirected lines (—) between entity set and relationship set represent signifying “many”.

- Relationship between **User**, **Gender**, **Vocation**, and **Religion**

Entity set **User** has many-to-one relationship with entity **Gender**, **Vocation**, and **Religion** respectively. Sine

one user only can choose one gender type, one vocation type and one religion type.

- Relationship between **User** and **User_Follower**
It has many-to-many relationship. Since each user can follow many users. And one user also can be followed by many users.

- Relationship between **User**, **Groups** and **Group_Member**
The two entity sets **User** and **Groups** are connected by the relationship set **Group_Member**. The entity set **User** and **Groups** have many-to-many relationship. Since one user can join many groups. And each group can have many group members.

- Relationship between **User**, **Topic** and **Topic_Follower**
The two entity sets **User** and **Topic** are connected by the relationship set **Topic_Follower**. The entity set The entity set **User** and **Topic** have many-to-many relationship. Since one user can follow many topics. And each topic can be followed by many users.

- Relationship between **Topic** and **Sub_Topic**
It has many-to-many relationship. Since each topic can have many sub topics. And a topic can be the sub topic of many topics.

-Relationship between **Post**, **Post_Image**, and P**ost_Link**
Entity set **Post** has one-to-many relationship with entity **Post_Image** and P**ost_Link** respectively. Since one post can include many images and links. But one specific image or link only belongs to one post.

- Relationship between **User** and **Post**
Entity sets **User** and **Post** have one-to-many relationship. Since one user can create many post. But one post only can belong to one user.

- Relationship between **User**, **Post**, and **Read_Post**
The two entity sets **User** and **Post** are connected by the relationship set **Read_Post**. The entity set **User** and **Post** have a many-to-many relationship. Since one user can read many posts. And one post can be read by many users. This relationship used to implement the function that a person is able to determine what posts have been added by people and/or topics that they are following since they last read from those people/topics. The post (postID) which has been read by the user(userID) will be added into **Read_Post** table. This can help program tracking the post that has been read by a specific user.

- Relationship between **User**, **Post**, **Simple_Respond_Type**, and **Simple_Respond**
The three entity set **User, Post** and **Simple_Respond_Type** are connected by the relationship set **Simple_Respond**. The three entity set have ternary relationship. The user can give a simple respond to a post by chose one simple respond type (i.e. Thumbs Up, Thumbs Down…).

- Relationship between **Topic**, **Post** and **Post_Topic**
The two entity set **Topic** and **Post** are connected by the relationship set **Post_Topic**. The entity set **Topic** and **Post** have many-to-many relationship. Since one post can belong to many topics. And one topic can have many posts.

- Relationship between **Post** and **Post_Respond**
It has one-to-many relationship. Since one post can have many respond post. But one respond post only belongs to one post.