Attributes

The attributes that need to be stored in our databases are:

User_id:

UUID
Char[32 characters]
The unique number that individualized a user's identity on the app.

User_firstname:

Char [64 characters]
The first name of the user.

User_lastname:

Char [64 characters]
The last name of the user .

User_email:

Char [64 characters]
The email of the user, the email associated with the user account, allows us to email to notify them.

User_password:

Char [32 characters]
The password of the user.

User_handle:

Char [32 characters]
The users handle that will appear on the as them on the app. (Screen name)

User_Status:

Char [16 characters]
The status of the user whether their account is approved, blacklisted, or paused.

Session_id:

UUID
Char [32 characters]
The ID of the current session the user is in on the application .

Owner_id:

UUID
Char[32 characters]
The unique number that is used to identify every group owner.

Post_ID:

UUID
Char[32 characters]
The unique number that is used to identify every post the user posts.

Poster_ID:

UUID
Char[32 characters]
The unique number that is used to identify every poster.

Post_title:

Char [64 characters]
The title of the post the user creates.

Post_content:

VarChar [0 to ~65500 characters]
The content of the post from the user could include the caption of the post, title of the post, etc.

Post_image:

Link
VarChar [0 - ~655000 characters]
The image of the post, the picture part of the post.

Post_likes_score:

Int [-2147483648 to 2147483647]
To avoid totalling every single like when someone pulls up their feed. Total of likes (+1 each).

Post_comments_count:

Int [-2147483648 to 2147483647]
Updated when new comments come in. (simplifies calculating post weight)

Post_visible:

Bool [0 indicates false, any nonzero integer indicates true]
The status of the post, meaning if the post is up and running, or if the post is hidden.

Reaction:

(+1 is currently the only value)

Group_ID:

UUID
Char [32 characters]

The unique ID number of the group so that we can tell the differences between the groups.

Group_name:
Char [64 characters]
The name of the group.

Group_ranking:
Integer [1-5]
An integer between 1 and 5 (inclusive) showing the user's preference for seeing the group at the top of the feed. (Higher values indicate a group to show earlier.)

Friend_id:
UUID
Char [32 characters]
The ID of user that the user_id is "friends" with.

Friendship_Status:
Enum
{nominal, accepted, pending, blocked, ..}
The states of if you are friends or not friends with the user or blocked / unblock etc.

Comment_ID:
UUID
Char [32 characters]
the unique ID that identifies the comment that was made on the post.

Commenter_ID:
UUID
Char[32 characters]
The unique number that is used to identify every commenter.

Comment_coment:
VarChar [0 to ~65500 characters]
The content of the comment.

Comment_timestamp:
Timestamp is a data type - see w3schools.
The time that the comment was made by a user, example being:
1985-09-25 17:45:30.005

Request_timestamp:
Timestamp is a data type - see w3schools.

The time that the friend request was made by a user, example being:
1985-09-25 17:45:30.005

Post_weight:
Integer [0 - max]
The calculated value of the post, used for generating an ordered feed for each user.

Functional Dependencies:

1. {user_id} -> {user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture}

2. {session_id} -> {user_id}

3. {post_id} -> {poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp}

4. {post_id, user_id} -> {reaction}

5. {group_id} -> {group_name, owner_id, group_ranking}

6. {user_id, friend_id} -> {friendship_status, request_timestamp}

7. {comment_id} -> {post_id, commenter_id, comment_content, comment_timestamp}

8. {user_id, post_id} -> {post_weight}

9. {post_id, group_id} -> {post_id, group_id} [by self-determination]

10. {group_id, user_id} -> {group_id, user_id} [by self-determination]

Normalization of the data model

The initial relvar is R{user_id, user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture, session_id, post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}, and satisfies all the FDs above.

The candidate key for R is {user_id, session_id, post_id, group_id, friend_id, comment_id} since {user_id, session_id, post_id, group_id, friend_id, comment_id}+ includes all the attributes of R, and no proper subset of {user_id, session_id, post_id, group_id, friend_id, comment_id} has a closure that includes all the attributes of R.

R is in 1NF, because the values of the attributes are atomic. R is not in 2NF, because we have attributes dependent only on a subset of the full candidate key, for instance, user_firstname is dependent on only user_id, not the full candidate key.

We apply Heath's Theorem to R, using FD 9.
A is {post_id, group_id}
B is {post_id, group_id}
C is { user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}

R1 is post_id, group_id, which satisfies FD 9. It is in at least 1NF, as it is derived from R, which is in 1NF. Since post_id and group_id depend only post_id and group_id non-transitively, and {post_id, group_id} is the candidate key, it is in 3NF. It satisfies FD 9.

R2 is {user_id, user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture, session_id, post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}, and satisfies all FDs 1-8 and 10. The candidate key is {user_id, session_id, post_id, group_id, friend_id, comment_id}. Since some attributes do not depend on the full candidate key (i.e. post_weight depends only on post_id and user_id), R2 is not in 2NF or higher. R2 satisfies FDs 1-8 and FD 10.

We apply Heath's Theorem to R2, using FD10.
A is {group_id, user_id}
B is {group_id, user_id}
C is { user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture, session_id, post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}

R2.1 is {group_id, user_id}. The candidate key is {group_id, user_id}. R2.1 is in 3NF, because there are no attributes other than the candidate key, so no attributes depend non-transitively on the candidate key or on a subset of the candidate key. It satisfies FD 10.

R2.2 is {user_id, user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture, session_id, post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}. It satisfies FDs 1-8.

The candidate key for R2.2 is {user_id, session_id, post_id, group_id, friend_id, comment_id}. R2.2 is in 1NF because R was in 1NF. Since some attributes do not depend on the full candidate key (i.e. post_weight depends only on post_id and user_id), R2.2 is not in 2NF or higher.

We apply Heath's Theorem using FD1 to decompose R2.2. Thus,
A is {user_id}
B is {user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture}
C is {session_id, post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}

R2.2.1 is: {user_id, user_firstname, user_lastname, user_email, user_password, user_handle, user_status, user_picture} and satisfies FD1. R2.2.1 is in at least 1NF, since it was produced by decomposing R. User_id is the candidate key, and all the attributes in R1 except user_id depend only on user_id, thus R2.2.1 is in at least 2NF. All non-key attributes are directly (non-transitively) dependent on user_id (the candidate key), so R2.2.1 is in 3NF.

R2.2.2 is: {user_id, session_id, post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, reaction, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}. R2.2.2 satisfies FDs 2-8. Its candidate key is {session_id, post_id, user_id, group_id, friend_id, comment_id}. The closure of that candidate key is all attributes in R2.2.2. R2.2.2 satisfies FDs 2-8.

R2.2.2 is in 1NF, since R was. It cannot be in 2NF, because (for example), post_content is dependent only on post_id, not on the full candidate key given above.

Next, we use Heath's theorem with FD4 to normalize R2.2.2:
A: {post_id, user_id}
B: {reaction}
C: {session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, group_id, group_name, owner_id,

group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}

R2.2.2.1 is {post_id, user_id, reaction} and satisfies FD4. R2.2.2.1's candidate key {post_id, user_id}. It is in at least 1NF because R2.2.2 was in 1NF. It is in at least 2NF because reaction only depends on the candidate key. It is in 3NF because this is a non-transitive relationship. This relvar satisfies FD 4.

R2.2.2.2 is {post_id, user_id, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp, post_weight}
R2.2.2.2's candidate key {session_id, post_id, group_id, user_id, friend_id, comment_id}.
R2.2.2.2 is in only 1NF, because user_id depends only on session_id, not the full candidate key. This relvar satisfives FDs 2-3, 5-8.

Apply FD8 to R2.2.2.2:
A: {user_id, post_id}
B: {post_weight}
C: {session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp}

R2.2.2.2.1 is {user_id, post_id, post_weight}. It satisfies FD8. R2.2.2.2.1's candidate key {user_id, post_id}. It is in 3NF because all attributes are atomic and only depend on user_id and post_id, the candidate key. It satisfies FD8.

R2.2.2.2.2 is {user_id, post_id, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, group_id, group_name, owner_id, group_ranking, friend_id, friendship_status, request_timestamp, comment_id, commenter_id, comment_content, comment_timestamp}.
R2.2.2.2.2's candidate key is {session_id, post_id, group_id, user_id, friend_id, comment_id}. It is in 1NF only because user_id depends only on session_id, not the full candidate key. It satisfies FDs 2, 3, 5, 6, and 7.

Apply FD6 to R2.2.2.2.2:
A: {user_id, friend_id}
B: {friendship_status, request_timestamp}
C: { post_id, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, group_id, group_name, owner_id, group_ranking, comment_id, commenter_id, comment_content, comment_timestamp}

R2.2.2.2.2.1 is {user_id, friend_id, friendship_status, request_timestamp}. R2.2.2.2.2.1's candidate key is {user_id, friend_id}. It is in 3NF because all attributes depend non-transitively on {user_id, friend_id}, the candidate key. It satisfies FD6.

R2.2.2.2.2.2 is {user_id, friend_id, post_id, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, group_id, group_name, owner_id, group_ranking, comment_id, commenter_id, comment_content, comment_timestamp}.
R2.2.2.2.2.2 candidate key {session_id, post_id, group_id, comment_id}. It satisfies FD 2, 3, 5, and 7.

Applying FD5 to R2.2.2.2.2.2:
A: {group_id}
B: {group_name, owner_id, group_ranking}
C: {user_id, friend_id, post_id, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, poster_id, comment_id, commenter_id, comment_content, comment_timestamp}

R2.2.2.2.2.2.1 is {group_id, group_name, owner_id, group_ranking}.
R2.2.2.2.2.2.1's candidate key is {group_id}. R2.2.2.2.2.2.1 is in 3NF because all other attributes depend non-transitively on group_id, the candidate key. It satisfies FD 5.

R2.2.2.2.2.2.2 is {group_id, user_id, friend_id, post_id, session_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp, comment_id, commenter_id, comment_content, comment_timestamp}
R2.2.2.2.2.2.2's candidate key is {session_id, post_id, comment_id}. This relvar satisfives FDs 2, 3, and 7.

Applying FD3 to R2.2.2.2.2.2.2:
A: {post_id}
B: {poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp}
C: {group_id, user_id, friend_id, session_id, comment_id, commenter_id, comment_content, comment_timestamp}

R2.2.2.2.2.2.2.1 is {post_id, poster_id, post_title, post_content, post_image, post_likes_score, post_comments_count, post_visible, post_timestamp}. R2.2.2.2.2.2.2.1's candidate key is {post_id}. It is in 3NF because all other attributes depend non-transitively on the candidate key. It satisfies FD 3.

R2.2.2.2.2.2.2.2 is {post_id, group_id, user_id, friend_id, session_id, comment_id, commenter_id, comment_content, comment_timestamp}. R2.2.2.2.2.2.2.2's candidate key is {session_id, comment_id}. It satisfies FDs 2 and 7.

Applying FD7 to R2.2.2.2.2.2.2.2.2:
A: {comment_id}
B: {post_id, commenter_id, comment_content, comment_timestamp}
C: {user_id, group_id, friend_id, session_id}

R2.2.2.2.2.2.2.2.2.1 is {comment_id, post_id, commenter_id, comment_content, comment_timestamp}. It satisfies FD7.
R2.2.2.2.2.2.2.2.2.1's candidate key is {comment_id}

R2.2.2.2.2.2.2.2.2.2 is {comment_id, group_id, user_id, friend_id, session_id}
R2.2.2.2.2.2.2.2.2.2's candidate key is {session_id}. It satisfies FD 2.

Applying FD2 to R2.2.2.2.2.2.2.2.2.2:
A: {session_id}
B: {user_id}
C: {comment_id, friend_id, group_id}

R2.2.2.2.2.2.2.2.2.2.1 is {session_id, user_id}. It satisfies FD2.
R2.2.2.2.2.2.2.2.2.2.1's candidate key is {session_id}. This relvar is in 3NF, because it is atomic, the attribute other than session_id depends only on the candidate key, and the relationship is non-transitive.

R2.2.2.2.2.2.2.2.2.2.2 is {session_id, comment_id, friend_id, group_id}
R2.2.2.2.2.2.2.2.2.2.2 has no candidate key, and satisfies no functional dependencies.

The final database design is below. All functional dependencies are satisfied and no information has been lost.

Final database design

The final database design is:
(primary keys = underlined, foreign keys = italics)

Table: Post_groups (shows which groups can see a specific post) [R1]
post_id, group_id

Table: Group_memberships (shows users in each group) [R2.1]
group_id, user_id

Table: Users
user_id, user_firstname, user_lastname, user_email, user_password, user_handle, user_status

Table: Posts
post_id, *poster_id*, post_title, post_content, post_image, post_likes_score,

post_comments_count, post_visible, post_timestamp

Table: Reactions
post_id, *user_id*, reaction

Table: Groups
group_id, group_name, *owner_id*, group_ranking

Table: Friendships
user_id, *friend_id*, friendship_status, request_timestamp

Table: Comments
comment_id, *post_id*, *commenter_id*, comment_content, comment_timestamp

Table: Sessions
session_id, *user_id*

Table: Posts_feed
user_id, *post_id*, post_weight