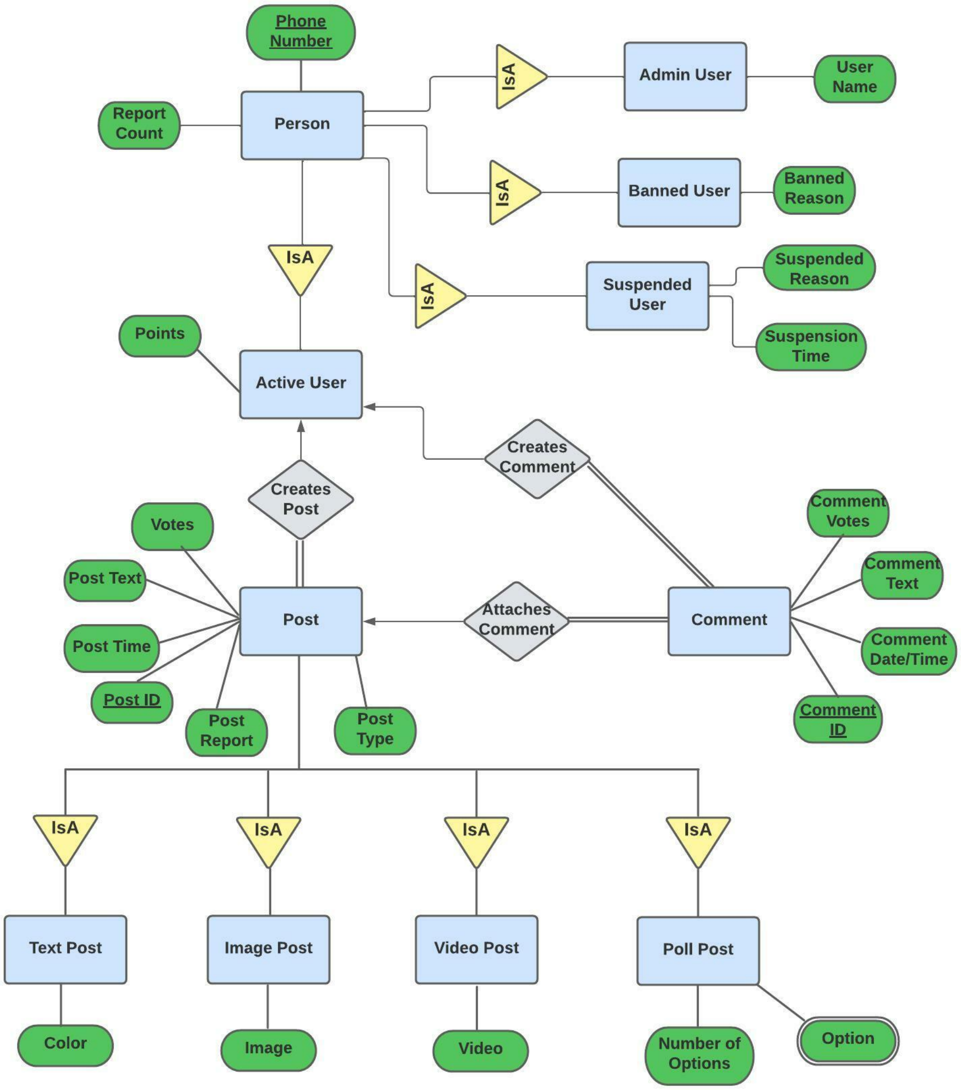
Michael Yates, mjy5xy
Justin Logan, jhl3mn

A reference to our code can be found at: https://github.com/Mike-Yates/Milestone2

Updated Diagram:
To Post: we added “Post Type” attribute. We also chose to delete reports from Comment.



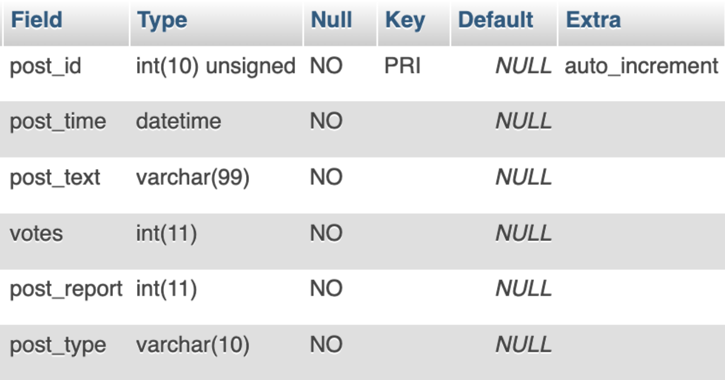
### 1) Database server

We are hosting our Database on the CS servers. It is currently part of the database assigned to us (named after my computing ID).

## 2) Table Schema

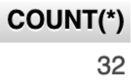
Screenshots of Table Schema:

**Post:** Post(post_id, post_time, post_text, votes, post_report, post_type)



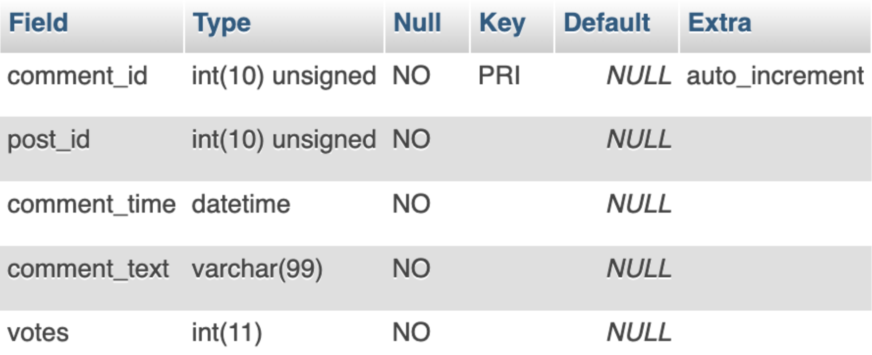
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| post_id | int(10) unsigned | NO | PRI | *NULL* | auto_increment |
| post_time | datetime | NO | | *NULL* | |
| post_text | varchar(99) | NO | | *NULL* | |
| votes | int(11) | NO | | *NULL* | |
| post_report | int(11) | NO | | *NULL* | |
| post_type | varchar(10) | NO | | *NULL* | |

Rows:



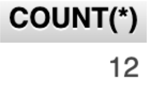
| COUNT(*) |
|---|
| 32 |

**Comment:** Comment(comment_id, post_id, comment_time, comment_text, votes)



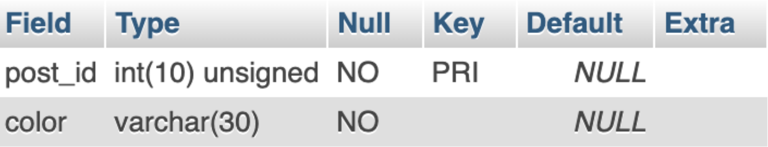
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| comment_id | int(10) unsigned | NO | PRI | *NULL* | auto_increment |
| post_id | int(10) unsigned | NO | | *NULL* | |
| comment_time | datetime | NO | | *NULL* | |
| comment_text | varchar(99) | NO | | *NULL* | |
| votes | int(11) | NO | | *NULL* | |

Rows:



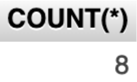
| COUNT(*) |
|---|
| 12 |

**Text_Post:** Text_Post(post_id, color)



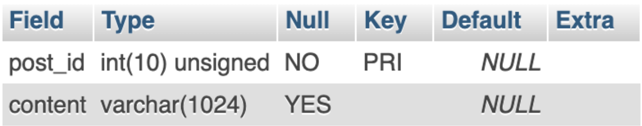
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| post_id | int(10) unsigned | NO | PRI | *NULL* | |
| color | varchar(30) | NO | | *NULL* | |

Rows:



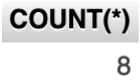
| COUNT(*) |
|---|
| 8 |

**Images:** Images(post_id, content)



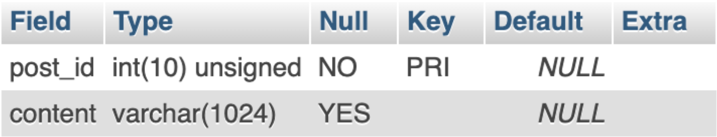
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| post_id | int(10) unsigned | NO | PRI | *NULL* | |
| content | varchar(1024) | YES | | *NULL* | |

Rows:



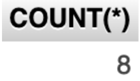
| COUNT(*) |
|---|
| 8 |

**Videos:** Videos(post_id, content)



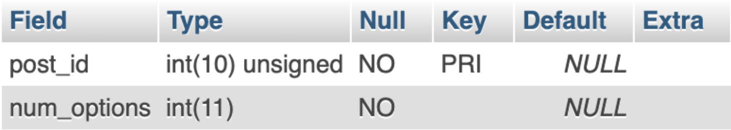
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| post_id | int(10) unsigned | NO | PRI | *NULL* | |
| content | varchar(1024) | YES | | *NULL* | |

Rows:



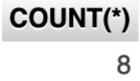
| COUNT(*) |
|---|
| 8 |

**Poll_Post:** Poll_Post(post_id, num_options)



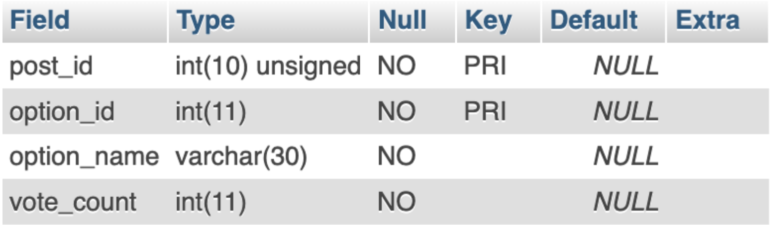
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| post_id | int(10) unsigned | NO | PRI | *NULL* | |
| num_options | int(11) | NO | | *NULL* | |

Rows:



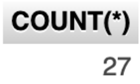
| COUNT(*) |
|---|
| 8 |

**Poll_Options:** Poll_options(post_id, option_id, option_name, vote_count)



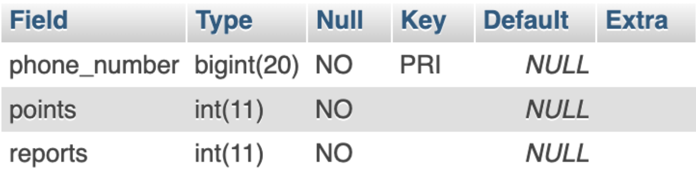
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| post_id | int(10) unsigned | NO | PRI | *NULL* | |
| option_id | int(11) | NO | PRI | *NULL* | |
| option_name | varchar(30) | NO | | *NULL* | |
| vote_count | int(11) | NO | | *NULL* | |

Rows:



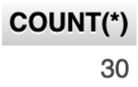
| COUNT(*) |
|---|
| 27 |

**Active_User:** ActiveUser(phone_number, points, reports)



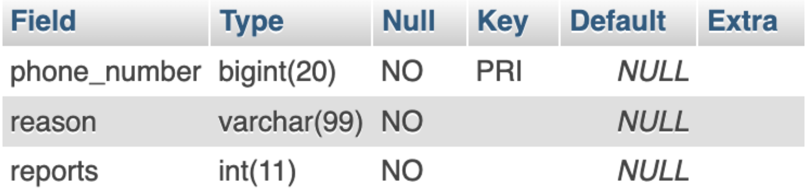
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| phone_number | bigint(20) | NO | PRI | *NULL* | |
| points | int(11) | NO | | *NULL* | |
| reports | int(11) | NO | | *NULL* | |

Rows:



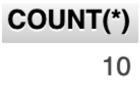
| COUNT(*) |
|---|
| 30 |

**Banned_User:** BannedUser(phon_number, reason, reports)



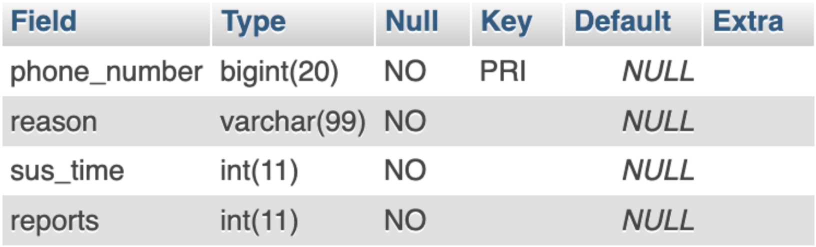
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| phone_number | bigint(20) | NO | PRI | *NULL* | |
| reason | varchar(99) | NO | | *NULL* | |
| reports | int(11) | NO | | *NULL* | |

Rows:



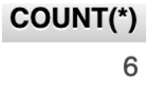
| COUNT(*) |
|---|
| 10 |

**Suspended_User:** SuspendedUser(phone_number, reason, sus_time, reports)



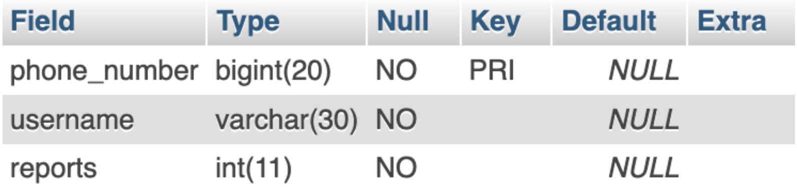
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| phone_number | bigint(20) | NO | PRI | *NULL* | |
| reason | varchar(99) | NO | | *NULL* | |
| sus_time | int(11) | NO | | *NULL* | |
| reports | int(11) | NO | | *NULL* | |

Rows:



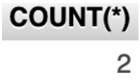
| COUNT(*) |
|---|
| 6 |

**Admin_User:** AdminUser(phone_number, username, reports)



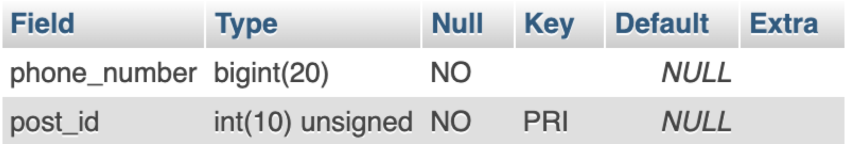
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| phone_number | bigint(20) | NO | PRI | *NULL* | |
| username | varchar(30) | NO | | *NULL* | |
| reports | int(11) | NO | | *NULL* | |

Rows:



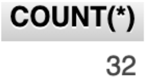
| COUNT(*) |
|---|
| 2 |

**Post_Creator:** Post_Creator(post_id, phone_number)



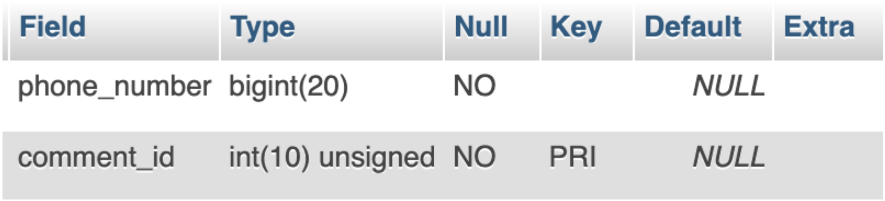
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| phone_number | bigint(20) | NO | | *NULL* | |
| post_id | int(10) unsigned | NO | PRI | *NULL* | |

Rows:



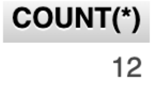
| COUNT(*) |
|---|
| 32 |

**Comment_Creator:** Comment_Creator(comment_id, phone_number)



| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| phone_number | bigint(20) | NO | | *NULL* | |
| comment_id | int(10) unsigned | NO | PRI | *NULL* | |

Rows:



| COUNT(*) |
|---|
| 12 |

Tables: As suggested in our Milestone Feedback 1, we decided to add a “post type” feature to our Post database. This lets us add a post database to store all common features of the 4 types of posts.

1. Post(post_id, post_time, post_text, votes, post_report, post_type)
2. Comment(comment_id, post_id, comment_time, comment_text, votes)
3. Text_Post(post_id, color)
4. Images(post_id, content)
5. Videos(post_id, content)
6. Poll_Post(post_id, num_options)
7. Poll_Options(post_id, option_id, option_name, vote_count)
8. Active_User(phone_number, points, reports)
9. Banned_User(phone_number, reason, reports)
10. Suspended_User(phone_number, reason, sus_time, reports)
11. Admin_User(phone_number, username, reports)
12. Post_Creator(post_id, phone_number)
13. Comment_Creator(comment_id, phone_number)

**Updated: Decompose tables using 3NF or BCNF and remove reflexivity,**

1. {post_id->post_time, post_text, votes, post_report, post_type}
2. {comment_id->post_id, comment_time, comment_text, votes}
3. {post_id->color}
4. {post_id->content}
5. {post_id->content}
6. {post_id->num_options}
7. {post_id, option_id->option_name, vote_count}
8. {phone_number->points, reports}
9. {phone_number->reason, reports}
10. {phone_number->reason, sus_time, reports}
11. {phone_number->username, reports}

Make sure we aren’t missing any dependencies (transitivity).
Confirm the following is a minimal basis:

The left hand side (LHS) of all of the dependencies (other than #7) cannot have any attributes removed without losing dependency because they are all of length one. With regard to the seventh dependency {post_id, option_id->option_name, vote_count}, if either attribute on the left side was removed then we would not be able to determine the vote count of that poll option for that specific poll.
Conclusion: this is a minimal basis.

Find Fc (the canonical cover)

- There are no trivial FDs. No Reflexivity. No Extraneous attributes.
- To verify it is in 3NF, we must show **lossless join** and **dependency preserving**.

R1 = post_id, post_time, post_text, votes, post_report, post_type
R2 = comment_id, post_id, comment_time, comment_text, votes
R3 = post_id, color
R4 = post_id, content
R5 = post_id, content
R6 = post_id, num_options
R7 = post_id, option_id, option_name, vote_count
R8 = phone_number, points, reports
R9 = phone_number, reason, reports
R10 = phone_number, reason, sus_time, reports
R11 = phone_number, username, reports

Lossless Join:
The superkey of the tables are post_id, phone_number, comment_id, option_id

- There is no overlap within the tables, so joining the tables would not increase the number of rows.

Dependency preserving:
For every non-trivial FD, X -> Attribute(s), X is a superkey.

- Every non-key attribute of each table only implies itself. The dependency is preserved.

Our tables meet the requirements of 3NF

### 3) Describe, justify, and provide evidence showing that your data are practical and sufficient

The Dataset we provided was made to represent an accurate usage of the app shortly after launch. The hope is for the app to catch on and accumulate hundreds to thousands of users with thousands of posts. For now, it was more worthwhile to input data when the usage is still relatively small as it likely will be for a while after launch. To simulate this we created 30 Active users with 2 admin users (Justin and Mike). The number of admin users will always be small and for the foreseeable future will only be Justin and Mike. The 30 active users have created 32 total posts. We predict that most users will not post significantly as is the case with any social media app so many of our users have no posts to their name. There are 6 suspended users and 10 banned users. Banned users will be a constantly increasing number as more users violate the rules of the app but for now it will be relatively small. Of the 32 posts, 8 are text, image, video, and polls respectively. This is to show the unique features of the app and the 4 different types of posts available. The poll posts have 27 total poll options spread between the 8 posts. This is to reflect the flexibility of the poll feature. Polls can have a large variety of poll options depending on the question being asked. Depending on how the app grows, it is likely that certain posts will become more popular than others but for now we cannot predict which that would be. We also have 12 comments created by active users attached to 10 different posts. Not every post will spark a comment and some posts will inspire more comments than others. Overall we believe this dataset adequately demonstrates the main functionality of our app and simulates what the database will look like shortly after launch when populated with real users. The specifics of the dataset we provided can be found in our SQL file.

### 4) List of all SQL commands used in your app

We wrote template commands for each of the following requests that will be made in our app:

1. AddComment
2. AddPollPost
3. AddPollOption
4. AddTextPost
5. AddImagePost
6. AddVideoPost
7. AddActiveUser
8. AddAdmin
9. DeleteComment
10. DeletePollPost
11. DeleteTextPost
12. DeleteImagePost
13. DeleteVideoPost
14. DeleteActiveUser
15. DeleteAdmin
16. UpdateUserReports
17. UpdatePostReports
18. UpdatePostLikes
19. UpdatePollOptionLikes
20. UpdateUserScore
21. GetPostDatetime
22. GetPostCreator
23. GetPostText
24. GetPostReportNumber
25. GetPostType
26. GetCommentCreator
27. GetPollOptions
28. GetTextPostColor

29. GetImagePost
30. GetVideoPost
31. GetUserScore
32. SortPostsByLikes

```
-- AddComment
INSERT INTO 'Comment' VALUES (<comment_id>, <post_id>, <comment_time>,
<comment_text>, <votes>);
INSERT INTO 'Comment_Creator' VALUES (<comment_id>, <phone_number>);

-- AddPollPost
INSERT INTO 'Post' VALUES (<post_id>, <post_time>, <post_text>, <post_text>,
<post_report>, 'poll');
INSERT INTO 'Post_Creator' VALUES (<phone_number>, <post_id>);
INSERT INTO 'Poll_Post' VALUES (<post_id>, <num_options>);

-- AddPollOption
INSERT INTO 'Poll_Options' VALUES (<post_id>, <option_id>, <option_name>, <vote_count>);

-- AddTextPost
INSERT INTO 'Post' VALUES (<post_id>, <post_time>, <post_text>, <post_text>,
<post_report>, 'text');
INSERT INTO 'Post_Creator' VALUES (<post_id>, <phone_number>);
INSERT INTO 'Text_Post' VALUES (<post_id>, <color>);

-- AddImagePost
INSERT INTO 'Post' VALUES (<post_id>, <post_time>, <post_text>, <post_text>,
<post_report>, 'image');
INSERT INTO 'Post_Creator' VALUES (<post_id>, <phone_number>);
INSERT INTO 'Images' VALUES (<post_id>, <content>);

-- AddVideoPost
INSERT INTO 'Post' VALUES (<post_id>, <post_time>, <post_text>, <post_text>,
<post_report>, 'video');
INSERT INTO 'Post_Creator' VALUES (<post_id>, <phone_number>);
INSERT INTO 'Videos' VALUES (<post_id>, <content>);

-- AddActiveUser
INSERT INTO 'Active_User' VALUES (<phone_number>, <points>, <reports>);

-- AddAdmin
INSERT INTO 'Admin_User' VALUES (<phone_number>, <username>, <reports>);

-- DeleteComment
```

```
DELETE FROM `Comment` WHERE `comment_id`=<comment_id>;

-- DeletePollPost
DELETE FROM 'Post' WHERE `post_id`=<post_id>;
DELETE FROM 'Post_Creator' WHERE `post_id`=<post_id>;
DELETE FROM 'Poll_Post' WHERE `post_id`=<post_id>;

DELETE FROM 'Poll_Options' WHERE `post_id`=<post_id>;

-- DeleteTextPost
DELETE FROM 'Post' WHERE `post_id`=<post_id>;
DELETE FROM 'Post_Creator' WHERE `post_id`=<post_id>;
DELETE FROM 'Text_Post' WHERE `post_id`=<post_id>;

-- DeleteImagePost
DELETE FROM 'Post' WHERE `post_id`=<post_id>;
DELETE FROM 'Post_Creator' WHERE `post_id`=<post_id>;
DELETE FROM 'Images' WHERE `post_id`=<post_id>;

-- DeleteVideoPost
DELETE FROM 'Post' WHERE `post_id`=<post_id>;
DELETE FROM 'Post_Creator' WHERE `post_id`=<post_id>;
DELETE FROM 'Videos' WHERE `post_id`=<post_id>;

-- DeleteActiveUser
DELETE FROM 'Active_User' WHERE `phone_number`=<phone_number>;

-- DeleteAdmin
DELETE FROM 'Admin_User' WHERE `phone_number`=<phone_number>;

-- UpdateUserReports
UPDATE 'Active_User' SET 'reports' = <reports> WHERE 'phone_number' = <phone_number>;

-- UpdatePostReports
UPDATE 'Post' SET 'reports' = <reports> WHERE 'post_id' = <post_id>;

-- UpdatePostLikes
UPDATE 'Post' SET 'votes' = <votes> WHERE 'post_id' = <post_id>;

-- UpdatePollOptionLikes
UPDATE 'Poll_Options' SET 'vote_count' = <vote_count> WHERE 'post_id' = <post_id> &&
'option_id'=<option_id>;

-- UpdateUserScore
```

```
UPDATE 'Active_User' SET 'points' = <points> WHERE 'phone_number' = <phone_number>;

-- GetPostDatetime
SELECT `post_time` FROM `Post` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- GetPostCreator
SELECT 'phone_number' from 'Post_Creator' WHERE 'post_id' = <post_id>;

-- GetPostText
SELECT `post_text` FROM `Post` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- GetPostReportNumber
SELECT `post_report` FROM `Post` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- GetPostType
SELECT `post_type` FROM `Post` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- GetCommentCreator
SELECT 'phone_number' from 'Comment_Creator' WHERE 'post_id' = <post_id>;

-- GetPollOptions
SELECT 'option_name' from 'Poll_Options' WHERE 'post_id' = <post_id> && 'option_id' = <option_id>;

-- GetTextPostColor
SELECT 'color' from 'Text_Post' WHERE 'post_id' = <post_id>;

-- GetImagePost
SELECT `content` FROM `Images` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- GetVideoPost
SELECT `content` FROM `Videos` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- GetUserScore
SELECT `points` FROM `Active_User` WHERE 'post_id' = <post_id>; -- change post id accordingly

-- SortPostsByLikes
SELECT * FROM `Post` ORDER BY `votes` DESC
```