CSCI 585 – Homework 2

1. List the ids and names of users who have no posts and have one or more comments on POST_ID=5.

```sql
SELECT U.USER_ID, U.NAME
FROM USERS U JOIN COMMENTS C
ON U.USER_ID = C.COMMENTER_USER_ID
WHERE C.POST_ID = 5
AND U.USER_ID NOT IN (
    SELECT USER_ID FROM POSTS
);
```

Since we are listing users with no posts, the user won't have an entry in POSTS table. Hence, we put a condition that user id should not be present in POSTS.USER_ID set. Also, they have one or more comment on post 5, we check for records with a COMMENTER_USER_ID and POST_ID = 5.

2. List the USER_ID of female mutual friends between users 1 and 2.

```sql
SELECT USERS.USER_ID
FROM USERS JOIN FRIENDSHIPS
ON FRIENDSHIPS.FRIEND_ID = USERS.USER_ID
WHERE FRIENDSHIPS.USER_ID = 1
AND USERS.USER_ID IN (
    SELECT USERS.USER_ID
    FROM USERS JOIN FRIENDSHIPS
    ON FRIENDSHIPS.FRIEND_ID = USERS.USER_ID
    WHERE FRIENDSHIPS.USER_ID = 2
)
AND GENDER = "F";
```

First, we join USERS and FRIENDSHIPS on FRIEND_ID column so that we get records corresponding to friends of users (In this case, users 1 and 2). We check ids of friends of one user into set of ids of friends of another user. Finally, we add a constraint saying that friend's gender should be "Female".

3. List the USER_ID of users who have more than 2 friends whom have at least one post

```sql
SELECT U.USER_ID
FROM USERS U JOIN FRIENDSHIPS F
ON U.USER_ID = F.USER_ID
WHERE F.FRIEND_ID IN (
    SELECT DISTINCT USR.USER_ID
    FROM USERS USR JOIN POSTS P
    ON USR.USER_ID = P.USER_ID
)
GROUP BY U.USER_ID
HAVING COUNT(F.FRIEND_ID) > 2;
```

First, we find the set of users having at least 1 post (Having entry in POSTS table). Then we count such users (Who are friends of users that we need). Finally, we put condition that count of such users should be more than 2.

4. List unique USER_ID of female users who were born after ‘1990-12-20’ and commented on posts of USER_ID=10. Show their friends count in a separate column.

```
SELECT U.USER_ID, COUNT(F.FRIEND_ID) AS NO_OF_FRIENDS
FROM USERS U JOIN FRIENDSHIPS F
ON U.USER_ID = F.USER_ID
WHERE U.USER_ID IN (
    SELECT USER_ID
    FROM USERS
    WHERE GENDER="F"
    AND DATE_OF_BIRTH > "1990-12-20"
)
AND U.USER_ID IN (
    SELECT COMMENTER_USER_ID
    FROM POSTS P JOIN COMMENTS C
    ON P.POST_ID = C.POST_ID
    WHERE P.USER_ID = 10
)
GROUP BY U.USER_ID;
```

First, we get set of users with gender “F” and born after the given date and set of all users who have commented on post by user 10. We mandate that the user we are looking for should be in both these sets. After that we count their friends and display that in a separate column.

5. List the USER_ID of users who commented on POST_ID=7 and are friends with the post creator.

```
SELECT F.FRIEND_ID AS USER_ID
FROM USERS U JOIN FRIENDSHIPS F JOIN POSTS P
ON U.USER_ID = F.USER_ID
AND U.USER_ID = P.USER_ID
WHERE P.POST_ID = 7
AND F.FRIEND_ID IN (
    SELECT C.COMMENTER_USER_ID
    FROM COMMENTS C JOIN POSTS P
    ON C.POST_ID = P.POST_ID
    WHERE P.POST_ID = 7
);
```

First, we join USERS and POSTS so that when POST_ID is set to 7, USERS.USER_ID contains id of the post creator. Then we look for his friends in set of all commenters who commented on post 7.

6. List the USER_ID and NAME of the 3 most female commenters, who are friends with USER_ID=20, with at least 3 comments on all the posts combined, excluding the comments under ones posted by USER_ID=10 and themselves. Show their augmented count of comments in a separate column. Also, show their total number of comments in another separate column.

```
SELECT U.USER_ID, U.NAME, TOTAL_SET.AUGMENTED_COUNT_OF_COMMENTS,
TOTAL_SET.TOTAL_COUNT_OF_COMMENTS
FROM (
    SELECT ACC_SET.COMMENTER_USER_ID, ACC_SET.AUGMENTED_COUNT_OF_COMMENTS,
COUNT(ACC_SET.COMMENTER_USER_ID) AS TOTAL_COUNT_OF_COMMENTS
    FROM (
        SELECT FF_20_C.COMMENTER_USER_ID, COUNT(FF_20_C.COMMENTER_USER_ID) AS
AUGMENTED_COUNT_OF_COMMENTS
        FROM(
            SELECT POST_ID, COMMENTER_USER_ID
            FROM COMMENTS C JOIN (
                SELECT U.USER_ID, U.NAME
                FROM USERS U JOIN FRIENDSHIPS F
                ON U.USER_ID = F.FRIEND_ID
                WHERE F.USER_ID = 20 AND U.GENDER="F"
            ) FF_20
            ON FF_20.USER_ID = C.COMMENTER_USER_ID) FF_20_C
        JOIN POSTS P
        ON FF_20_C.POST_ID = P.POST_ID
        WHERE P.USER_ID != 10 AND P.USER_ID NOT IN (
            SELECT COMMENTER_USER_ID
            FROM (
                SELECT POST_ID, COMMENTER_USER_ID
                FROM COMMENTS C JOIN (
                    SELECT U.USER_ID, U.NAME
                    FROM USERS U JOIN FRIENDSHIPS F
                    ON U.USER_ID = F.FRIEND_ID
                    WHERE F.USER_ID = 20 AND U.GENDER="F"
                ) FF_20
                ON FF_20.USER_ID = C.COMMENTER_USER_ID
            ) FF_20_C
        )
        GROUP BY FF_20_C.COMMENTER_USER_ID
        HAVING AUGMENTED_COUNT_OF_COMMENTS > 2
        ORDER BY AUGMENTED_COUNT_OF_COMMENTS DESC
    ) ACC_SET JOIN COMMENTS C
    ON ACC_SET.COMMENTER_USER_ID = C.COMMENTER_USER_ID
    GROUP BY ACC_SET.COMMENTER_USER_ID
) TOTAL_SET JOIN USERS U
ON TOTAL_SET.COMMENTER_USER_ID = U.USER_ID
ORDER BY TOTAL_SET.AUGMENTED_COUNT_OF_COMMENTS DESC
LIMIT 3;
```

First, we find all the set of all female friends of user 20 in set FF_20 (Female Friends of 20). Then we inner join this set with comments to find which of these friends are also commenters and get set FF_20_C. Then we join this set with posts to find which one of these commenters commented on posts by user id 10 and by themselves (That's why same query which gave us FF_20_C set is repeated). In this new set (ACC_SET), we count the no. of comments, with conditions that no. of comments be greater than 2 (at least 3). This gives us augmented count of comments. Note that we order these records with descending order of ACC, so that when we take top 3 records later, we get users with max ACC. Now we join this set (ACC_SET) with Comments again, to find total count of comments (As there are no constraints here like post id or minimum number of comments, all the records are counted) that is represented by TOTAL. Finally, we join with USERS table so that we can get names of these users.