CSCI-585 Database Systems
Submitted By: Amit Kumar
USC ID : 7088752227

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

Solution:

```
SELECT USER_ID, NAME FROM USERS WHERE USER_ID IN (SELECT
DISTINCT USERS.USER_ID FROM USERS  LEFT JOIN POSTS ON
USERS.USER_ID = POSTS.USER_ID WHERE POSTS.POST_ID IS NULL)
AND USER_ID IN (SELECT USERS.USER_ID FROM USERS LEFT JOIN
COMMENTS ON USERS.USER_ID = COMMENTS.COMMENTER_USER_ID WHERE
COMMENTS.COMMENT_ID IS NOT NULL AND POST_ID=5);
```

Explanation:

1. This query has two sub-queries.
2. The first sub-query does a left outer join of the USER table and POSTS table using the column USER_ID with the condition that it returns only the rows where the POST_ID is null. This means this will return all the rows where the POST_ID is null. After that I have used the DISTINCT keyword to select only the distinct USERS. So the final output of this sub-query is IDs of the USERS which do not have a single POST in their name.
3. The second sub-query does a left outer join of the USERS table and COMMENTS table using the USER_ID column in the USERS table and COMMENTER_USER_ID in the COMMENTS table as the common column. It also contain two conditions that the COMMENT_ID should not be null and the POST_ID should be equal to 5. So the final output of this sub-query will be the IDs of the USERS which have commented on the POST_ID=5.
4. The complete query is selecting the USER_IDs and NAMEs of the USERS whose USER_ID is part of the results returned by the two sub-queries explained in point 2 and 3.

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

Solution:

```
SELECT USER_ID FROM USERS WHERE USER_ID IN (SELECT USER_ID
FROM USERS WHERE USER_ID IN (SELECT FRIEND_ID FROM
FRIENDSHIPS WHERE USER_ID=1) AND GENDER='F') AND USER_ID IN
(SELECT USER_ID FROM USERS WHERE USER_ID IN (SELECT USER_ID
FROM USERS WHERE USER_ID IN (SELECT FRIEND_ID FROM
FRIENDSHIPS WHERE USER_ID=2) AND GENDER='F'));
```

Explanation:

1. This query is in the following form:

   main-query (sub-query-1 (sub-query-2)) AND (sub-query-3 (sub-query-4))

   Where:

   sub-query-2 => SELECT FRIEND_ID FROM FRIENDSHIPS WHERE USER_ID=1

   sub-query-1 => SELECT USER_ID FROM USERS WHERE USER_ID IN (**sub-query-2**) AND GENDER=‘F’

   sub-query-4 => SELECT FRIEND_ID FROM FRIENDSHIPS WHERE USER_ID=2

   sub-query-3 => SELECT USER_ID FROM USERS WHERE USER_ID IN (**sub-query-4**) AND GENDER=‘F’

   main-query => SELECT USER_ID FROM USERS WHERE USER_ID IN (**sub-query-1**) AND USER_ID IN (**sub-query-3**)

2. The sub-query-2 gets the FRIEND_IDs of all the users which are friends to the USER_ID=1.

3. The sub-query-1 selects the female USER_IDs from the USER_IDs returned by the sub-query-2(explained in point 2).

4. The sub-query-4 gets the FRIEND_IDs of all the users which are friends to the USER_ID=2.

5. The sub-query-3 selects the female USER_IDs from the USER_IDs returned by the sub-query-4(explained in point 4).

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

Solution:

```
SELECT FRIENDSHIPS.USER_ID FROM FRIENDSHIPS WHERE FRIEND_ID
IN (SELECT DISTINCT(USER_ID) FROM POSTS) GROUP BY USER_ID
HAVING COUNT(*) > 2;
```

Explanation:

1. This query is in the following form.

    main-query (sub-query-1)

2. The sub-query-1 fetches the USER_IDs of the users which have at-least one post in the POSTS table.
3. The main-query selects the users from the FRIENDSHIPS table where the FRIEND_ID is in the result returned by the sub-query-1(explained in point 2 above) and groups the result returned by the USER_ID and returns only the rows which have had COUNT(*) greater than 2. Here Count(*) represents the count of the friends each user had.

===

Question 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.

Solution:

```
SELECT U.USER_ID, COUNT(*) FROM USERS U INNER JOIN
FRIENDSHIPS F USING(USER_ID) WHERE U.GENDER='F' AND
U.DATE_OF_BIRTH > '1990-12-20' AND U.USER_ID IN (SELECT
DISTINCT(C.COMMENTER_USER_ID) FROM POSTS P INNER JOIN
COMMENTS C USING(POST_ID) WHERE P.USER_ID=10) GROUP BY
U.USER_ID;
```

Explanation:

1. This query has can be divided into 2 separate parts.
2. The first part gets the distinct USER_IDs of the users who commented on the POSTs by USER_ID=10 by doing an inner join on the COMMENTS table and POSTS table.

3. The second part does an inner join on the USERS table and FRIENDSHIPS table and selects only the USER_IDs explained in point 2. As this join will have multiple rows for a single user because a user can have many friends, so it does a group by USER_ID and displays the USER_ID and count of records in each group.

==

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

Solution:

```
SELECT COMMENTER_USER_ID FROM COMMENTS WHERE POST_ID=7 AND
EXISTS (SELECT * FROM FRIENDSHIPS WHERE USER_ID =(SELECT
USER_ID FROM POSTS WHERE POST_ID=7) AND
FRIEND_ID=COMMENTER_USER_ID);
```

Explanation:

1. This query has can be divided into 3 separate parts.
2. The first part gets the USER_ID of the user who created the post with POST_ID=7.
3. The second part gets the friends of the USER who created the POST_ID=7.
4. The third part selects the COMMENTER_USER_ID from the COMMENTS table whose IDs are equal to the FRIEND_ID returned in the step explained in point 3.

==

Question 6:

Solution:

==