**Database Assignment No. 2**

**SQL**

**Name: Rahul Lele USC ID: 2483165273**

QUERY 1

Question: List the ids and names of users who have no posts and have one or more

comments on POST\_ID=5.

SELECT USER\_ID, NAME FROM USERS

WHERE USER\_ID NOT IN (SELECT USER\_ID FROM POSTS)

AND USER\_ID IN (SELECT COMMENTER\_USER\_ID FROM COMMENTS WHERE POST\_ID=5);

Explanation:

Here, the user-ids and names of users are selected from the users table on two conditions. The conditions in ‘WHERE’ are connected by ‘AND’ means both the conditions must be true. The first condition is a ‘NOT IN’ operator with subquery. The subquery returns the all the user\_ids who have made posts. The ‘NOT IN’ together with the subquery ensure that the user\_id selected from users table has no posts. The second condition is a ‘IN’ operator with subquery. The subquery returns all the user\_ids who have made comments on post\_id=5. The ‘IN’ together with the subquery ensure that the user\_id selected from users table has made comments on post\_id=5.

QUERY 2

Question: List the USER\_ID of female mutual friends between user 1 and 2.

SELECT USER\_ID FROM USERS

WHERE USER\_ID IN (SELECT FRIEND\_ID FROM FRIENDSHIPS WHERE USER\_ID=1)

AND USER\_ID IN (SELECT FRIEND\_ID FROM FRIENDSHIPS WHERE USER\_ID=2) AND GENDER='F';

Explanation:

Here, the user-ids of users are selected from the users table on three conditions. The conditions in ‘WHERE’ are connected by ‘AND’ means both all the three conditions must be true. The first condition is a ‘IN’ operator with subquery. The subquery returns all the user\_ids who are friends of user\_id = 1. The ‘IN’ together with the subquery ensure that the user\_id selected from users table is a friend of user\_id=2. The second condition is a ‘IN’ operator with subquery. The subquery returns all the user\_ids who are friends of user\_id = 2. The ‘IN’ together with the subquery ensure that the user\_id selected from users table is a friend of user\_id=2. The third condition ‘GENDER=F’ ensures the user selected is a female.

QUERY 3

Question: List the USER\_ID of users who have more than 2 friends whom have at least one post.

SELECT FRIENDSHIPS.USER\_ID FROM

FRIENDSHIPS WHERE FRIENDSHIPS.FRIEND\_ID IN (SELECT POSTS.USER\_ID FROM POSTS)

GROUP BY FRIENDSHIPS.USER\_ID

HAVING COUNT(FRIENDSHIPS.FRIEND\_ID)>2;

Explanation:

The WHERE condition has an ‘IN’ operator with a subquery. The subquery returns all user\_id of users who have made at least one post. The subquery with ‘IN’ operator checks whether the friend of the selected user has made at least one post. For every user, we need to check whether, it has more than two friends whom have at least one post. So we group by user\_id on a condition that the user must have more than two friends.

QUERY 4

Question: 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 DISTINCT U1.USER\_ID,(SELECT COUNT(FRIEND\_ID)FROM FRIENDSHIPS WHERE FRIENDSHIPS.USER\_ID=U1.USER\_ID)AS FRIENDS

FROM (USERS U1 JOIN COMMENTS C ON U1.USER\_ID=C.COMMENTER\_USER\_ID) JOIN

(USERS U2 JOIN FRIENDSHIPS FR ON U2.USER\_ID=FR.USER\_ID)

ON U1.USER\_ID=U2.USER\_ID

WHERE C.POST\_ID IN (SELECT POSTS.POST\_ID FROM POSTS WHERE POSTS.USER\_ID=10) AND U1.DATE\_OF\_BIRTH>'1990-12-20'

AND U1.GENDER='F'

GROUP BY USER\_ID;

Explanation:

In order to show count of friends, the attribute is itself shown as a separate query which returns the count of friends for every user using the aggregate function ‘COUNT’. The attribute is given the alias ‘FRIENDS’. The users table is joined with friends table to get the comments made by each user. The user table is joined with friends table in order to get the friends every user has. Both these join tables are joined together. The whole purpose of this is to get the comments made by every user as well as the list of friends every user has. Now, to get the comments made by the user on posts by user\_id=10, we use the ‘IN’ operator and for every user we check if the comments made by the user are on posts by user-id=10. The other two conditions are date of birth after ‘1990-12-20’ and gender is female.

QUERY 5

Question: List the USER\_ID of users who commented on POST\_ID=7 and are friends with the post creater.

SELECT COMMENTER\_USER\_ID FROM COMMENTS

WHERE POST\_ID=7 AND COMMENTER\_USER\_ID IN (SELECT FRIEND\_ID FROM POSTS JOIN FRIENDSHIPS ON POSTS.USER\_ID = FRIENDSHIPS.USER\_ID

WHERE POST\_ID=7);

Explanation:

The outer query gives the commenters on post\_id=7. To check if they are friends with the creaters of post\_id=7, we have written the second condition in WHERE clause. The inner query joins posts with friendships which gives the friends of every post creater. The inner query returns the friends of post creater of post\_id=7. Then, we check if the commenter on post\_id=7 is in this friendlist or not.

QUERY 6

SELECT U1.USER\_ID, U1.NAME,COUNT(C1.COMMENT\_ID)

FROM (FRIENDSHIPS FR JOIN USERS U1 ON FR.FRIEND\_ID=U1.USER\_ID) JOIN COMMENTS C1 ON U1.USER\_ID=C1.COMMENTER\_USER\_ID

WHERE FR.USER\_ID=20 AND U1.GENDER='F' AND C1.COMMENT\_ID NOT IN (SELECT C2.COMMENT\_ID FROM POSTS P1 JOIN COMMENTS C2 ON P1.POST\_ID=C2.POST\_ID WHERE P1.USER\_ID=10 OR P1.USER\_ID=U1.USER\_ID) GROUP BY U1.USER\_ID

HAVING COUNT(C1.COMMENT\_ID)>=3

ORDER BY COUNT(C1.COMMENT\_ID) DESC

LIMIT 3;

Explanation:

The friendships table is joined with users table to get user details for friends of USER\_ID=20. This is joined with comments table to get the comments made by the friends of user\_id=20. There is a condition on what types comments should be considered. This is executed using the subquery in the join condition. The subquery returns comments on posts by user\_id=10 as well as comments on their own posts posted by friends of user\_id=20. The ‘NOT IN’ operator ensures that the join schema does not have comments on posts by user\_id=10 as well as comments on their own posts. Each of the selected female user must have at least three comments excluding the ones on posts by user\_id=10 and their group comments. This is done by the HAVING clause. We need to order the top 3 female commenters in descending order. This is done by the ORDER BY clause and LIMIT clause.