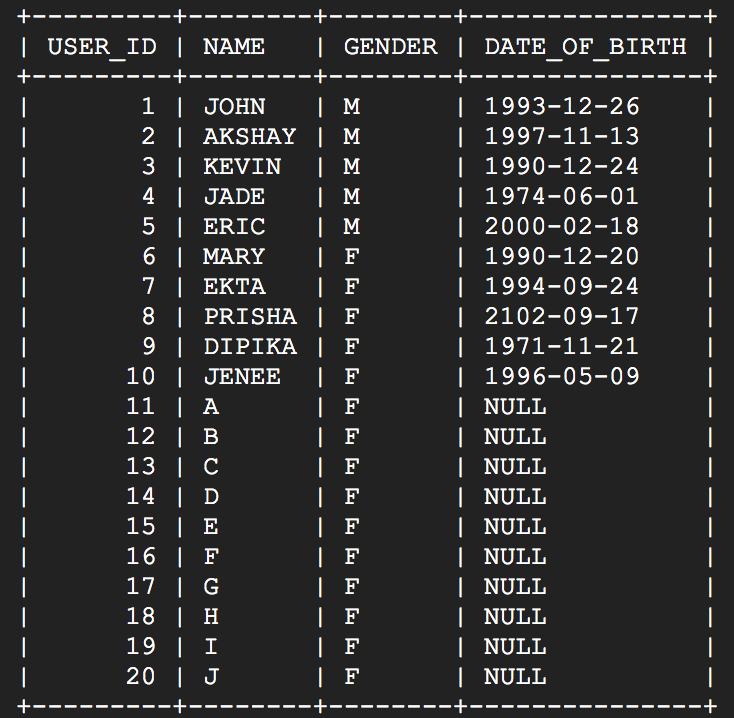
**Home Work-2**

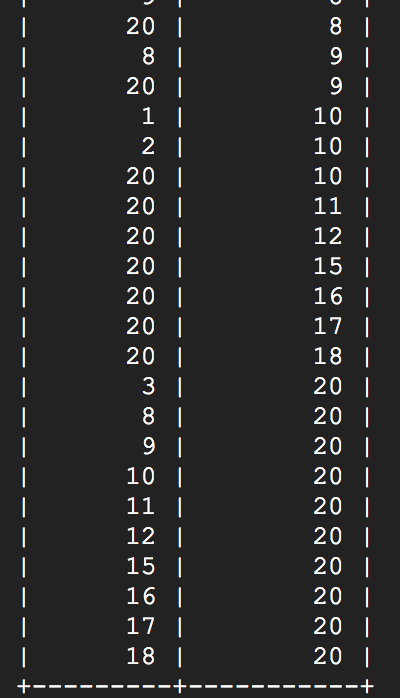
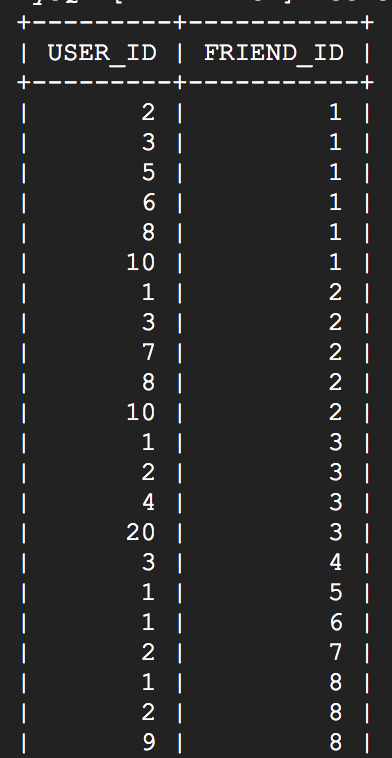
**Note: Nicely formatted query is in Ekta\_Rita\_HW2.docx file**

The tables used for the queries are as follows:

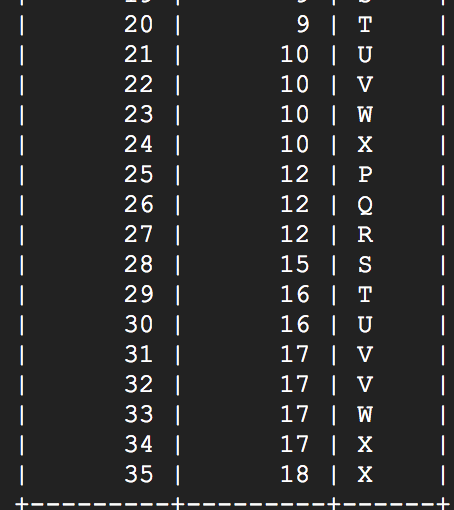
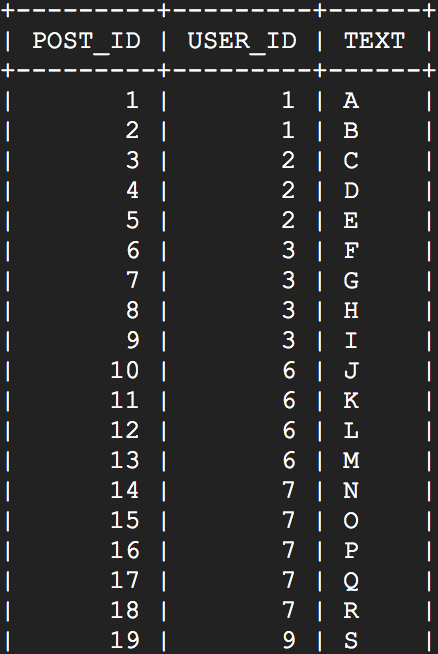
USERS TABLE



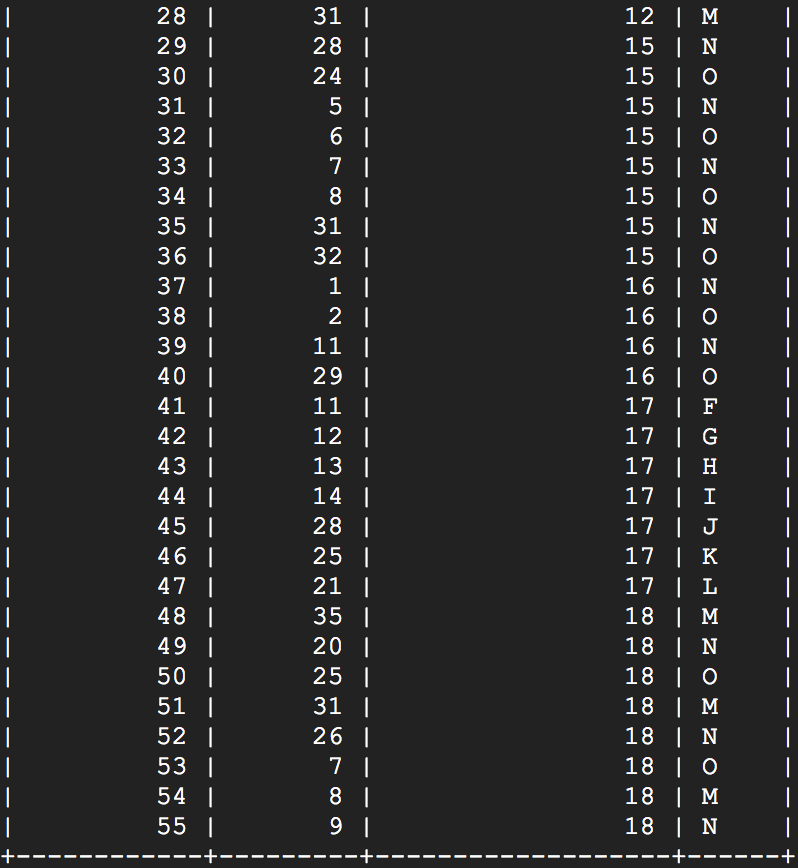
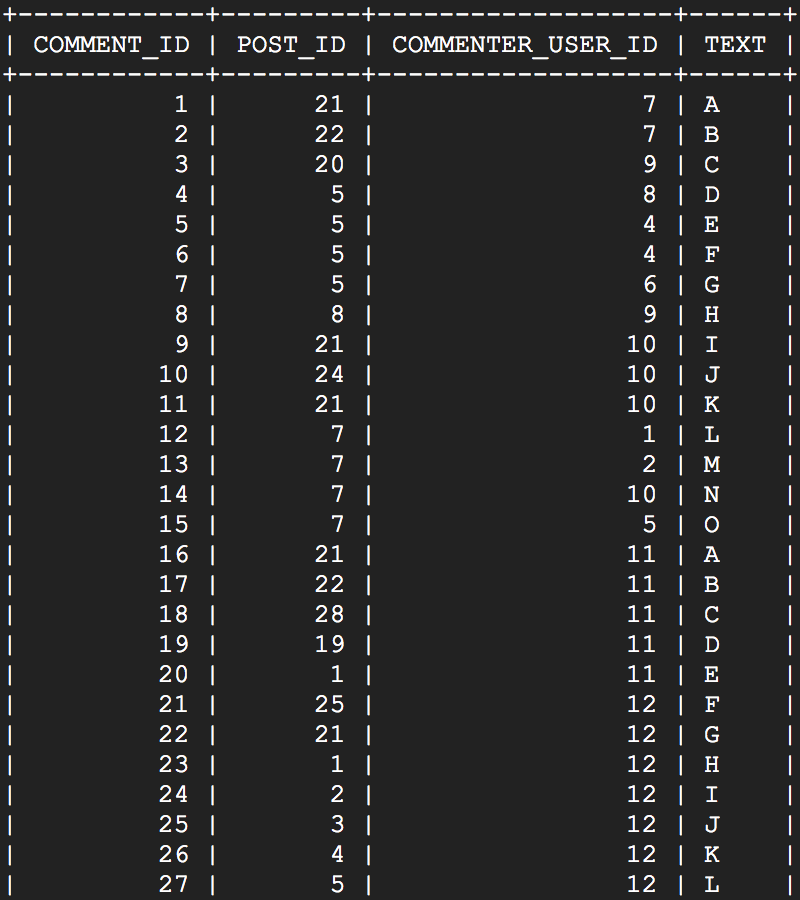
FRIENDSHIPS TABLE



POSTS TABLE



COMMENTS TABLE



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

SELECT DISTINCT U.USER\_ID, U.NAME

FROM COMMENTS C LEFT JOIN USERS U

ON C.COMMENTER\_USER\_ID = U.USER\_ID

WHERE C.POST\_ID = 5

AND U.USER\_ID NOT IN

(

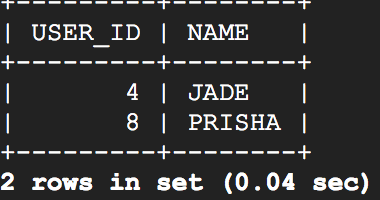
SELECT P.USER\_ID FROM POSTS P

)

ORDER BY U.USER\_ID;

**Explanation:**

The query selects the USER\_ID and NAME of those users who have commented on POST\_ID = 5 and which are not in the result returned by the inner query which finds the users who have at least 1 post. There is a left join in table COMMENTS and USERS so that it maps the COMMENTER\_USER\_ID to USER\_ID. The result obtained is as follows:



**2. List the USER\_ID of female mutual friends between users 1 and 2.**

SELECT F1.FRIEND\_ID MUTUAL\_FRIEND\_ID

FROM FRIENDSHIPS F1, USERS U1

WHERE F1.USER\_ID = 1

AND F1.FRIEND\_ID = U1.USER\_ID

AND U1.GENDER = 'F'

AND F1.FRIEND\_ID IN

(

SELECT F2.FRIEND\_ID

FROM FRIENDSHIPS F2, USERS U2

WHERE F2.USER\_ID = 2

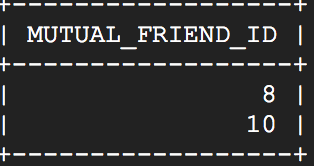
AND F2.FRIEND\_ID = U2.USER\_ID

AND U2.GENDER = 'F'

);

**Explanation:**

Finding all the female friends of USER\_ID = 1 and all the female friends of USER\_ID = 2. After this only those female friends of USER\_ID = 1 is selected who are also there in the female friend list of USER\_ID = 2. The result obtained is as follows:



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

SELECT F.USER\_ID

FROM FRIENDSHIPS F RIGHT JOIN

(

SELECT DISTINCT P1.USER\_ID FROM POSTS P1

) P2

ON F.FRIEND\_ID = P2.USER\_ID

GROUP BY F.USER\_ID

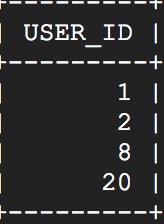
HAVING COUNT(F.FRIEND\_ID) > 2

ORDER BY F.USER\_ID;

**Explanation:**

Getting only those distinct friends who have posted at least 1 post from the POSTS table and counting these friends of each user by using a right join which maps the FRIEND\_ID to USER\_ID.

Now selecting only those users whose friend count from above condition is greater than 2 by using HAVING clause. Assuming more than 2 means greater than 2 and not greater than equal to 2. The result obtained is as follows:



**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 USR.USER\_ID, COUNT(F.FRIEND\_ID) FRIEND\_COUNT

FROM

(

SELECT DISTINCT U.USER\_ID FROM USERS U, COMMENTS C

WHERE U.GENDER = 'F' AND U.DATE\_OF\_BIRTH > '1990-12-20'

AND C.COMMENTER\_USER\_ID = U.USER\_ID

AND C.POST\_ID IN

(

SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

)

) USR LEFT JOIN FRIENDSHIPS F

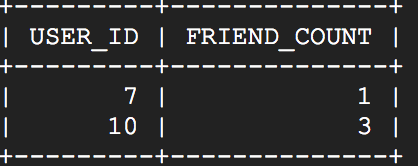
ON USR.USER\_ID = F.USER\_ID

GROUP BY USR.USER\_ID;

**Explanation:**

Select the USER\_ID of female users who satisfy the given condition and then left join it with the FRIENDSHIPS table to get the count of the friends of each user. The inner query selects distinct users because there can be duplicate users who have commented on multiple posts.

Assuming born after ‘1990-12-20’ does not include that date. The result obtained is as follows:



**5. List the USER\_ID of users who commented on POST\_ID=7 and are friends with the post creator.**

SELECT DISTINCT C.COMMENTER\_USER\_ID

FROM

(

SELECT F.FRIEND\_ID

FROM FRIENDSHIPS F

WHERE F.USER\_ID =

(

SELECT P.USER\_ID

FROM POSTS P

WHERE P.POST\_ID = 7

)

) PF

LEFT JOIN

COMMENTS C

ON C.COMMENTER\_USER\_ID = PF.FRIEND\_ID

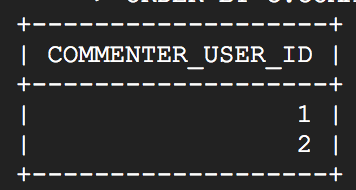
WHERE C.POST\_ID = 7

ORDER BY C.COMMENTER\_USER\_ID;

**Explanation:**

Get the list of friends of the user who has posted with POST\_ID = 7 and then join these friends with the COMMENTS table on their FRIEND\_ID and COMMENTER\_USER\_ID and the post on which they comment is POST\_ID = 7.

I have used DISTINCT C.COMMENTER\_USER\_ID because there is a possibility that the user must have commented more than once on POST\_ID = 7. The result obtained is as follows:



**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 F.USER\_ID, F.NAME, AUG.COUNT AUGMENTED\_COMMENTS\_COUNT, F.NON\_AUGMENTED\_TOTAL TOTAL\_COUNT

FROM

( SELECT FU.USER\_ID, FU.NAME,

COUNT(COMMENTS.COMMENT\_ID) NON\_AUGMENTED\_TOTAL

FROM COMMENTS

RIGHT JOIN

( SELECT U.USER\_ID, U.NAME

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20

AND U.GENDER = 'F'

) FU

ON COMMENTS.COMMENTER\_USER\_ID = FU.USER\_ID

GROUP BY FU.USER\_ID, FU.NAME

) F

RIGHT JOIN

( SELECT A.USER\_ID1 USER\_ID, A.COUNT1 COUNT

FROM

( SELECT AUG2.USER\_ID1, AUG2.COUNT1, AUG2.USER\_ID2, AUG2.COUNT2,

AUG2.USER\_ID3, COUNT(C.COMMENT\_ID) COUNT3,

AUG2.COUNT1 + AUG2.COUNT2 + COUNT(C.COMMENT\_ID) TOTAL

FROM

( SELECT AUG1.USER\_ID1, AUG1.COUNT1, AUG1.USER\_ID2,

COUNT(C.COMMENT\_ID) COUNT2, AUG1.USER\_ID3

FROM

( SELECT UC.USER\_ID1, COUNT(C.COMMENT\_ID) COUNT1,

UC.USER\_ID2, UC.USER\_ID3

FROM

( SELECT FU1.USER\_ID USER\_ID1, FU2.USER\_ID USER\_ID2,

FU3.USER\_ID USER\_ID3

FROM

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU1,

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU2,

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU3

WHERE FU1.USER\_ID < FU2.USER\_ID

AND FU2.USER\_ID < FU3.USER\_ID

ORDER BY FU1.USER\_ID, FU2.USER\_ID, FU3.USER\_ID

) UC, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = UC.USER\_ID1

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = UC.USER\_ID1

OR P.USER\_ID = UC.USER\_ID2

OR P.USER\_ID = UC.USER\_ID3

)

GROUP BY UC.USER\_ID1, UC.USER\_ID2, UC.USER\_ID3

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

) AUG1, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = AUG1.USER\_ID2

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = AUG1.USER\_ID1

OR P.USER\_ID = AUG1.USER\_ID2

OR P.USER\_ID = AUG1.USER\_ID3

)

GROUP BY AUG1.USER\_ID1, AUG1.USER\_ID2, AUG1.USER\_ID3

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

) AUG2, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = AUG2.USER\_ID3

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = AUG2.USER\_ID1

OR P.USER\_ID = AUG2.USER\_ID2

OR P.USER\_ID = AUG2.USER\_ID3

)

GROUP BY AUG2.USER\_ID1, AUG2.USER\_ID2, AUG2.USER\_ID3

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

ORDER BY

(AUG2.COUNT1 + AUG2.COUNT2 + COUNT(C.COMMENT\_ID)) DESC,

AUG2.COUNT1 DESC, AUG2.COUNT2 DESC, COUNT(C.COMMENT\_ID) DESC

LIMIT 1

) A

UNION

SELECT A.USER\_ID2, A.COUNT2

FROM

( SELECT AUG2.USER\_ID1, AUG2.COUNT1, AUG2.USER\_ID2, AUG2.COUNT2,

AUG2.USER\_ID3, COUNT(C.COMMENT\_ID) COUNT3,

AUG2.COUNT1 + AUG2.COUNT2 + COUNT(C.COMMENT\_ID) TOTAL

FROM

( SELECT AUG1.USER\_ID1, AUG1.COUNT1, AUG1.USER\_ID2,

COUNT(C.COMMENT\_ID) COUNT2, AUG1.USER\_ID3

FROM

( SELECT UC.USER\_ID1, COUNT(C.COMMENT\_ID) COUNT1,

UC.USER\_ID2, UC.USER\_ID3

FROM

( SELECT FU1.USER\_ID USER\_ID1, FU2.USER\_ID USER\_ID2,

FU3.USER\_ID USER\_ID3

FROM

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU1,

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU2,

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU3

WHERE FU1.USER\_ID < FU2.USER\_ID

AND FU2.USER\_ID < FU3.USER\_ID

ORDER BY FU1.USER\_ID, FU2.USER\_ID, FU3.USER\_ID

) UC, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = UC.USER\_ID1

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = UC.USER\_ID1

OR P.USER\_ID = UC.USER\_ID2

OR P.USER\_ID = UC.USER\_ID3

)

GROUP BY UC.USER\_ID1, UC.USER\_ID2, UC.USER\_ID3

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

) AUG1, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = AUG1.USER\_ID2

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = AUG1.USER\_ID1

OR P.USER\_ID = AUG1.USER\_ID2

OR P.USER\_ID = AUG1.USER\_ID3

)

GROUP BY AUG1.USER\_ID1, AUG1.USER\_ID2, AUG1.USER\_ID3

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

) AUG2, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = AUG2.USER\_ID3

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = AUG2.USER\_ID1

OR P.USER\_ID = AUG2.USER\_ID2

OR P.USER\_ID = AUG2.USER\_ID3

)

GROUP BY AUG2.USER\_ID1, AUG2.USER\_ID2, AUG2.USER\_ID3

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

ORDER BY

(AUG2.COUNT1 + AUG2.COUNT2 + COUNT(C.COMMENT\_ID)) DESC,

AUG2.COUNT1 DESC, AUG2.COUNT2 DESC, COUNT(C.COMMENT\_ID) DESC

LIMIT 1

) A

UNION

SELECT A.USER\_ID3, A.COUNT3

FROM

( SELECT AUG2.USER\_ID1, AUG2.COUNT1, AUG2.USER\_ID2, AUG2.COUNT2,

AUG2.USER\_ID3, COUNT(C.COMMENT\_ID) COUNT3,

AUG2.COUNT1 + AUG2.COUNT2 + COUNT(C.COMMENT\_ID) TOTAL

FROM

( SELECT AUG1.USER\_ID1, AUG1.COUNT1, AUG1.USER\_ID2,

COUNT(C.COMMENT\_ID) COUNT2, AUG1.USER\_ID3

FROM

( SELECT UC.USER\_ID1, COUNT(C.COMMENT\_ID) COUNT1,

UC.USER\_ID2, UC.USER\_ID3

FROM

( SELECT FU1.USER\_ID USER\_ID1, FU2.USER\_ID USER\_ID2,

FU3.USER\_ID USER\_ID3

FROM

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU1,

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU2,

( SELECT U.USER\_ID

FROM USERS U, FRIENDSHIPS F

WHERE U.USER\_ID = F.FRIEND\_ID

AND F.USER\_ID = 20 AND U.GENDER = 'F'

) FU3

WHERE FU1.USER\_ID < FU2.USER\_ID

AND FU2.USER\_ID < FU3.USER\_ID

ORDER BY FU1.USER\_ID, FU2.USER\_ID, FU3.USER\_ID

) UC, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = UC.USER\_ID1

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = UC.USER\_ID1

OR P.USER\_ID = UC.USER\_ID2

OR P.USER\_ID = UC.USER\_ID3

)

GROUP BY UC.USER\_ID1, UC.USER\_ID2, UC.USER\_ID3

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

) AUG1, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = AUG1.USER\_ID2

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = AUG1.USER\_ID1

OR P.USER\_ID = AUG1.USER\_ID2

OR P.USER\_ID = AUG1.USER\_ID3

)

GROUP BY AUG1.USER\_ID1, AUG1.USER\_ID2, AUG1.USER\_ID3

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

) AUG2, COMMENTS C

WHERE C.COMMENTER\_USER\_ID = AUG2.USER\_ID3

AND C.POST\_ID NOT IN

( SELECT P.POST\_ID

FROM POSTS P

WHERE P.USER\_ID = 10

OR P.USER\_ID = AUG2.USER\_ID1

OR P.USER\_ID = AUG2.USER\_ID2

OR P.USER\_ID = AUG2.USER\_ID3

)

GROUP BY AUG2.USER\_ID1, AUG2.USER\_ID2, AUG2.USER\_ID3

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

ORDER BY

(AUG2.COUNT1 + AUG2.COUNT2 + COUNT(C.COMMENT\_ID)) DESC,

AUG2.COUNT1 DESC, AUG2.COUNT2 DESC, COUNT(C.COMMENT\_ID) DESC

LIMIT 1

) A

) AUG

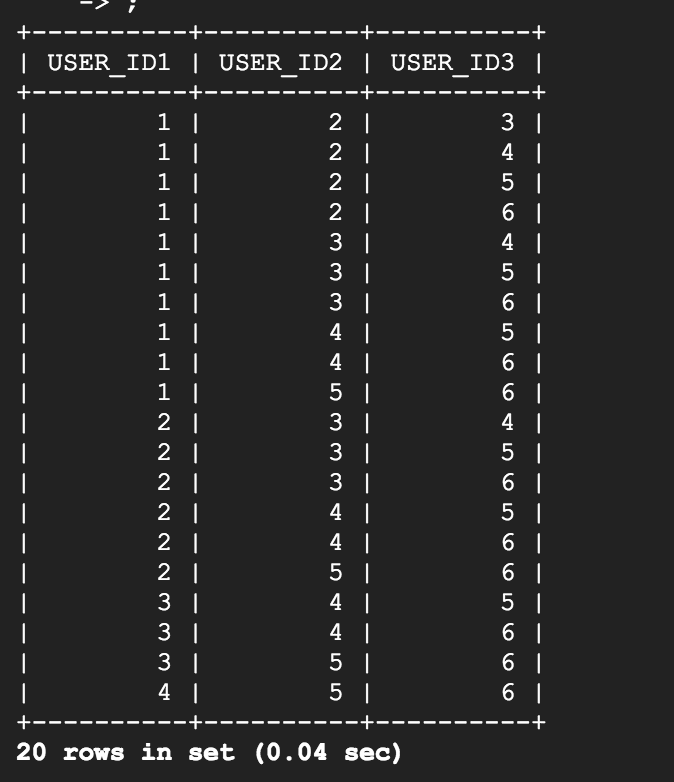
ON F.USER\_ID = AUG.USER\_ID

ORDER BY AUG.COUNT DESC, F.NAME;

**NOTE**: For this query I have used the data provided in the example. So the results obtained are for that data and not the above data.

**Explanation:**

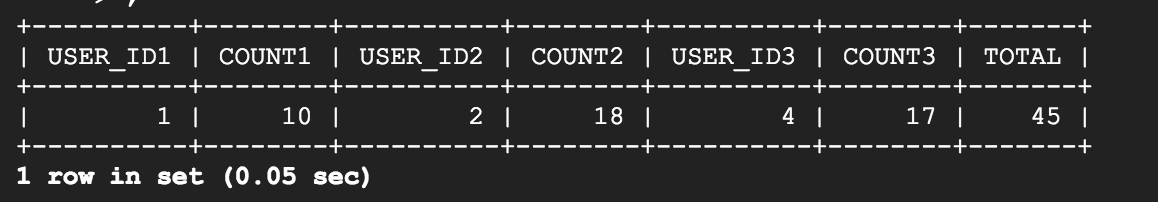
Firstly, find all the female users who are friends with USER\_ID = 20. After we get the list of all these female users (FU) find various combinations by joining this list twice where FU1.USER\_ID < FU2.USER\_ID and FU2.USER\_ID < FU3.USER\_ID. This will give the various triplets/user combinations (UC) that are possible in the following format:



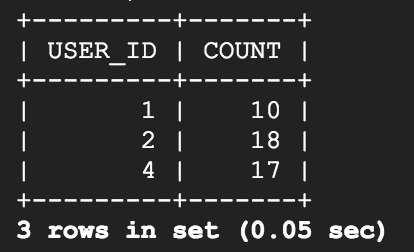
Now for each of these combinations find their augmented count of comments such that the count is >=3 starting with USER\_ID1, then the result obtained (AUG1) is used to find the augmented count of comments for USER\_ID2 and eventually this result (AUG2) is used for counting augmented comments for USER\_ID3. After getting the individual counts we find the sum of these counts for each combination and store it in the separate column.

We sort the result in descending order of the sum of augmented count of comments. If there is a tie, we sort it in the descending order of augmented count 1, then augmented count 2 and then augmented count 3.

The top most record is selected as a result which is given an alias name ‘A’. This intermediate table has the following structure:



Now we perform a union on this intermediate result by selecting USER\_ID1, COUNT1 first union with USER\_ID2, COUNT2 union with USER\_ID3, COUNT3. We get an intermediate result as follows which we give an alias name ‘AUG’:



Next, we find all the female users who are friends with USER\_ID = 20 along with their total count of comments on all the posts. This result is given an alias name ‘F’.

Finally, we perform a right join on F and AUG on F.USER\_ID = AUG.USER\_ID and sort them in the descending order of AUGMENTED\_COMMENTS\_COUNT and if there is a tie, we sort it in alphabetical order of NAME. The final result is as follows:

