**HW2**

Q1-  List the ids and names of users who have no posts and have one or more comments on POST\_ID=5

Ans:

SELECT USERS.USER\_ID, USERS.NAME FROM USERS

WHERE NOT EXISTS(SELECT POSTS.USER\_ID FROM POSTS)

AND EXISTS(SELECT COMMENTS.COMMENTER\_USER\_ID FROM COMMENTS

WHERE (COMMENTS.POST\_ID=5)

GROUP BY(COMMENTS.COMMENTER\_USER\_ID) HAVING COUNT(COMMENTS.COMMENT\_ID)>=1

AND USERS.USER\_ID=COMMENTS.COMMENTER\_USER\_ID);

[Explain]

Step1: “users who have one or more comments on POST\_ID=5” =>

using GROUP BY(COMMENTS.COMMENTER\_USER\_ID) to calculate how many comments posted by each user and using HAVING COUNT(COMMENTS.COMMENT\_ID) to limit the number of comments.

Step 2: combine Step1 and info from table USERS based on the description “List the ids and names of users who have no posts” => that is,

SELECT USERS.USER\_ID, USERS.NAME FROM USERS WHERE NOT EXISTS(SELECT POSTS.USER\_ID FROM POSTS) INTERSECT Step1

However, MySQL doesn’t have intersection operator, we can use EXISTS to simulate it.

For example,

SELECT contact\_id, last\_name, first\_name FROM contacts WHERE contact\_id < 100

INTERSECT SELECT customer\_id, last\_name, first\_name FROM customers WHERE last\_name <> 'Johnson';

It should be change into this form

SELECT contacts.contact\_id, contacts.last\_name, contacts.first\_name FROM contacts

WHERE contacts.contact\_id < 100

AND EXISTS (SELECT \* FROM customers WHERE customers.last\_name <> 'Johnson'

AND customers.customer\_id = contacts.contact\_id

AND customers.last\_name = contacts.last\_name

AND customers.first\_name = contacts.first\_name);

Q2-  List the USER\_ID of female mutual friends between users 1 and 2.

Ans:

SELECT t.USER\_ID FROM USERS AS t WHERE t.GENDER="F" AND t.USER\_ID IN

(SELECT t1.USER\_ID FROM FRIENDSHIPS AS t1 WHERE t1.FRIEND\_ID=1

AND t1.USER\_ID IN (SELECT t2.USER\_ID FROM FRIENDSHIPS AS t2 WHERE t2.FRIEND\_ID=2));

[Explain]

Step1: combine info from table FRIENDSHIPS based on the description: “mutual friends between users 1 and 2” => that is,

SELECT USER\_ID FROM FRIENDSHIPS WHERE FRIEND\_ID=1 INTERSECT SELECT USER\_ID FROM FRIENDSHIPS WHERE FRIEND\_ID=2;

However, MySQL doesn’t have intersection operator, we can use IN to simulate it.

For example,

SELECT category\_id FROM products INTERSECT SELECT category\_id FROM inventory;

It should be change into this form

SELECT products.category\_id FROM products WHERE products.category\_id IN (SELECT inventory.category\_id FROM inventory);

Therefore, we have the part

(SELECT t1.USER\_ID FROM FRIENDSHIPS AS t1 WHERE t1.FRIEND\_ID=1

AND t1.USER\_ID IN (SELECT t2.USER\_ID FROM FRIENDSHIPS AS t2 WHERE t2.FRIEND\_ID=2);

Step 2: combine info from table USERS and Step1 according to the description: “the USER\_ID of female …” => that is,

SELECT USER\_ID FROM USERS WHERE GENDER="F" INTERSECT result from Step1

Q3-  List the USER\_ID of users who have more than 2 friends whom have at least one post.

Ans:

SELECT DISTINCT USER\_ID FROM FRIENDSHIPS

WHERE EXISTS (SELECT DISTINCT USER\_ID FROM POSTS)

GROUP BY (USER\_ID) HAVING COUNT(FRIEND\_ID)>=2;

[Explain]

Step1: “people who have at least one post” (USER\_ID and FRIEND\_ID is a pair in FRIENDSHIPS)

SELECT DISTINCT USER\_ID FROM FRIENDSHIPS WHERE EXISTS (SELECT DISTINCT USER\_ID FROM POSTS)

Step2: “List the USER\_ID of users who have more than 2 friends”

SELECT USER\_ID FROM FRIENDSHIPS GROUP BY (USER\_ID) HAVING COUNT(FRIEND\_ID)>=2

Step3: combine Step1 & Step2

Q4-  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.

Ans:

SELECT FRIENDSHIPS.USER\_ID, COUNT(FRIENDSHIPS.FRIEND\_ID) FROM FRIENDSHIPS

WHERE FRIENDSHIPS.USER\_ID IN (

SELECT USERS.USER\_ID FROM USERS

WHERE (USERS.GENDER="F" AND USERS.DATE\_OF\_BIRTH > '1990-12-20')

AND EXISTS(

SELECT DISTINCT COMMENTS.COMMENTER\_USER\_ID FROM COMMENTS

WHERE COMMENTS.POST\_ID IN (SELECT POSTS.POST\_ID FROM POSTS

WHERE (POSTS.USER\_ID=10))

AND USERS.USER\_ID = COMMENTS.COMMENTER\_USER\_ID))

GROUP BY (FRIENDSHIPS.USER\_ID);

[Explain]

Step1: “posts of USER\_ID=10” => that is,

Result = SELECT POST\_ID FROM POSTS WHERE (USER\_ID=10)

Step2: “users commented on posts found in Step1”

ReVal = SELECT DISTINCT COMMENTER\_USER\_ID FROM COMMENTS WHERE (POST\_ID IN Result)

Step3: “USER\_ID of female users who were born after 1990-12-20”

Temp = SELECT USER\_ID FROM USERS WHERE GENDER="F" AND DATE\_OF\_BIRTH > '1990-12-20' AND EXISTS (ReVal AND USER\_ID = COMMENTER\_USER\_ID)

Step4: “Show their friends count in a separate column”

SELECT USER\_ID, COUNT(FRIEND\_ID) FROM FRIENDSHIPS WHERE (USER\_ID IN Temp)

Q5-  List the USER\_ID of users who commented on POST\_ID=7 and are friends with the post creator.

SELECT DISTINCT COMMENTS.COMMENTER\_USER\_ID FROM COMMENTS

WHERE (COMMENTS.POST\_ID=7) AND EXISTS(

SELECT DISTINCT FRIENDSHIPS.FRIEND\_ID FROM FRIENDSHIPS

WHERE (FRIENDSHIPS.USER\_ID IN

(SELECT DISTINCT POSTS.USER\_ID FROM POSTS WHERE(POSTS.POST\_ID=7)))

AND COMMENTS.COMMENTER\_USER\_ID = FRIENDSHIPS.FRIEND\_ID);

[Explain]

Step1: “the USER\_ID of users who commented on POST\_ID=7” => that is,

SELECT DISTINCT COMMENTS.COMMENTER\_USER\_ID FROM COMMENTS WHERE (COMMENTS.POST\_ID=1007)

Step2: “List the USER\_ID of users that are friends with the post (POST\_ID=7) creator.”

SELECT DISTINCT FRIENDSHIPS.FRIEND\_ID FROM FRIENDSHIPS WHERE (

FRIENDSHIPS.USER\_ID IN (SELECT DISTINCT POSTS.USER\_ID FROM POSTS WHERE(POSTS.POST\_ID=7)))

Step3: combine Step1 & Step2

Q6

Ans:……..Incomplete

SELECT COMMENTS.COMMENTER\_USER\_ID,COUNT(COMMENTS.COMMENT\_ID) AS ACC

FROM COMMENTS WHERE COMMENTS.COMMENT\_ID IN(

SELECT COMMENTS.COMMENT\_ID FROM POSTS

INNER JOIN COMMENTS ON (POSTS.POST\_ID = COMMENTS.POST\_ID) WHERE NOT

(COMMENTS.COMMENTER\_USER\_ID = POSTS.USER\_ID OR COMMENTER\_USER\_ID=10))

AND EXISTS(

SELECT USER\_ID, NAME FROM USERS WHERE (

GENDER="F" AND USER\_ID IN (SELECT USER\_ID FROM FRIENDSHIPS WHERE FRIEND\_ID=20))

AND USERS.USER\_ID = COMMENTS.COMMENTER\_USER\_ID)

GROUP BY (COMMENTS.COMMENTER\_USER\_ID) HAVING COUNT(COMMENTS.COMMENT\_ID)>=3

ORDER BY ACC DESC LIMIT 3;

[Explain]

1. USER is a Female:

SELECT USER\_ID, NAME FROM USERS WHERE GENDER="F"

2.USER has Friend 20:

SELECT USER\_ID FROM FRIENDSHIPS WHERE FRIEND\_ID=20

3. Comments do not post by USER itself & 10

SELECT COMMENTS.COMMENT\_ID,COMMENTS.COMMENTER\_USER\_ID,POSTS.USER\_ID FROM POSTS INNER JOIN COMMENTS ON (POSTS.POST\_ID = COMMENTS.POST\_ID) WHERE NOT (COMMENTS.COMMENTER\_USER\_ID = POSTS.USER\_ID OR COMMENTER\_USER\_ID=10)

4. at least 3 comments in Step3

SELECT COMMENTS.COMMENTER\_USER\_ID FROM COMMENTS WHERE COMMENTS.COMMENT\_ID IN(SELECT COMMENTS.COMMENT\_ID FROM POSTS INNER JOIN COMMENTS ON (POSTS.POST\_ID = COMMENTS.POST\_ID) WHERE NOT (COMMENTS.COMMENTER\_USER\_ID = POSTS.USER\_ID OR COMMENTER\_USER\_ID=10)) GROUP BY (COMMENTS.COMMENTER\_USER\_ID) HAVING COUNT(COMMENTS.COMMENT\_ID)>=3;

5. total comments

SELECT COMMENTS.COMMENTER\_USER\_ID,COUNT(COMMENTS.COMMENT\_ID) AS TOTAL FROM COMMENTS GROUP BY (COMMENTS.COMMENTER\_USER\_ID) HAVING COUNT(COMMENTS.COMMENT\_ID)>=3;