HW2 – SQL

Q1) List the ids and names of users who have no posts and have one or more comments on POST_ID=5.

- Sub Query result returns users who commented on POST_ID = 5.
- Joined the sub query result to user table as names of the users must be displayed.
- Then filtered the users to show only the users who did not post.

QUERY:

```
SELECT
  USERS.USER_ID,
  USERS.NAME
FROM (SELECT DISTINCT
  COMMENTER_USER_ID
FROM COMMENTS
WHERE POST_ID = 5) COMMG2
JOIN USERS
  ON COMMG2.COMMENTER_USER_ID = USERS.USER_ID
WHERE COMMG2.COMMENTER_USER_ID NOT IN (SELECT DISTINCT
  USER_ID
FROM POSTS);
```

Q2) List the USER_ID of female mutual friends between users 1 and 2.

- The innermost subquery returns users who are friends with USER_ID=2.
- Then filters users who are also friends with USER_ID = 1
- Later filters on GENDER to get only female users.
- Which gives the mutual female friends between users 1 and 2.

QUERY:

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

Q3) List the USER_ID of users who have more than 2 friends whom have at least one post.

- The sub query gives the users who have posted at least one post.
- Then from FRIENDSHIPS table we are retrieving all the users and their friends, filtering out the friends who have not posted anything.
- Then counting the USER_ID in the above resulting data who have more than 2 friends.
- Displaying only the USER_ID of the users who have more than 2 friends who posted.

QUERY:

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

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.

- The innermost sub query gives the POST_IDs OF USER_ID=10
- Then retrieve all the users who commented on posts of user 10.
- Filtered all the female users who are born after '1990-12-20', named this result as FEMALEUSERS.
- Now to display their friends joined the result FEMALEUSERS to FRIENDSHIPS table with a left join so that the users with no friends can also be displayed with friend count =0.

QUERY:

```
SELECT
  FEMALEUSERS.USER_ID,
  COUNT(FRIENDSHIPS.USER_ID)
FROM (SELECT
  USER_ID
FROM USERS
WHERE GENDER = 'F'
AND DATE_OF_BIRTH > '1990-12-20'
AND USER_ID IN (SELECT DISTINCT
  COMMENTER_USER_ID
FROM COMMENTS
WHERE POST_ID IN (SELECT
  POST_ID
FROM POSTS
WHERE POSTS.USER_ID = 10))) FEMALEUSERS
LEFT JOIN FRIENDSHIPS
  ON FEMALEUSERS.USER_ID = FRIENDSHIPS.USER_ID
GROUP BY FEMALEUSERS.USER_ID;
```

Q5) List the USER_ID of users who commented on POST_ID=7 and are friends with the post creator.
- The sub queries return the creator from POSTS table and list of user ids who commented on post 7.
- Now filtering out USER_IDs who are friends with post creator and also commented on POST_ID=7.

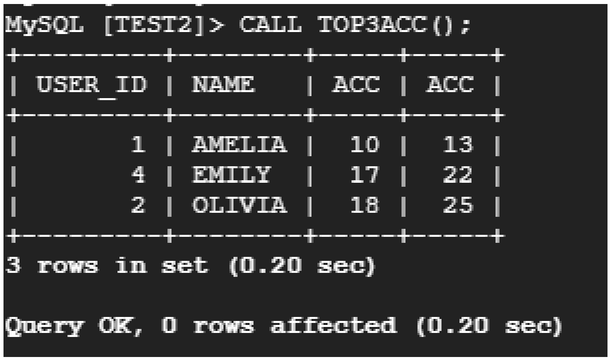
QUERY:

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

Q6)

- Have written a plsql block for this question.
- Where we get the result in ascending order of augmented count of comments.
- COLUMN 1 is the USER_ID.
- COLUMN 2 is the user name.
- COLUMN 3 is the augmented count of comments.
- COLUMN 4 is the total comments.
- Each user is displayed order of augmented count of comments. Please find below a sample output.
- ALSO ATTACHING .TXT FILE OF THE PROCEDURE NAMED **Q6.TXT**.
- The cursor gives all the triplets for users who are friends with USER_ID 20, commented at least 3 times without counting user_id 10 posts and self-posts.
- Then filtered out the maximum triplet from the cursor.
- Displayed each user with all the required data.

OUTPUT:



```
MySQL [TEST2]> CALL TOP3ACC();
+---------+--------+-----+-----+
| USER_ID | NAME   | ACC | ACC |
+---------+--------+-----+-----+
|       1 | AMELIA |  10 |  13 |
|       4 | EMILY  |  17 |  22 |
|       2 | OLIVIA |  18 |  25 |
+---------+--------+-----+-----+
3 rows in set (0.20 sec)

Query OK, 0 rows affected (0.20 sec)
```

PROCEDURE:

```
DELIMITER $$
CREATE PROCEDURE TOP3ACC()
BEGIN

DECLARE done INT DEFAULT 0;

DECLARE F1 INT DEFAULT NULL;
DECLARE F2 INT DEFAULT NULL;
DECLARE F3 INT DEFAULT NULL;

DECLARE F1_ACC INT DEFAULT NULL;
DECLARE F2_ACC INT DEFAULT NULL;
DECLARE F3_ACC INT DEFAULT NULL;

DECLARE F1_TOP3 INT DEFAULT -1;
DECLARE F2_TOP3 INT DEFAULT -1;
DECLARE F3_TOP3 INT DEFAULT -1;

DECLARE TOTAL INT DEFAULT 0;
DECLARE TEMP INT DEFAULT 0;
DECLARE EXIT1 INT DEFAULT 0;

DECLARE COMBINATIONS CURSOR FOR SELECT A.USER_ID, B.USER_ID, C.USER_ID FROM (SELECT
USERS.USER_ID FROM USERS JOIN FRIENDSHIPS ON USERS.USER_ID = FRIENDSHIPS.USER_ID JOIN
COMMENTS ON COMMENTS.COMMENTER_USER_ID = USERS.USER_ID WHERE USERS.GENDER='F'
AND FRIENDSHIPS.FRIEND_ID = 20 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS
WHERE USER_ID = 10 OR USER_ID = USERS.USER_ID)  GROUP BY COMMENTS.COMMENTER_USER_ID
HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3) A CROSS JOIN (SELECT USERS.USER_ID
FROM USERS JOIN FRIENDSHIPS ON USERS.USER_ID = FRIENDSHIPS.USER_ID JOIN COMMENTS ON
COMMENTS.COMMENTER_USER_ID = USERS.USER_ID WHERE USERS.GENDER='F' AND
FRIENDSHIPS.FRIEND_ID = 20 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS
WHERE USER_ID = 10 OR USER_ID = USERS.USER_ID)  GROUP BY COMMENTS.COMMENTER_USER_ID
HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3) B CROSS JOIN (SELECT USERS.USER_ID
FROM USERS JOIN FRIENDSHIPS ON USERS.USER_ID = FRIENDSHIPS.USER_ID JOIN COMMENTS ON
COMMENTS.COMMENTER_USER_ID = USERS.USER_ID WHERE USERS.GENDER='F' AND
FRIENDSHIPS.FRIEND_ID = 20 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS
WHERE USER_ID = 10 OR USER_ID = USERS.USER_ID)  GROUP BY COMMENTS.COMMENTER_USER_ID
HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3) C WHERE A.USER_ID != B.USER_ID AND
B.USER_ID != C.USER_ID AND C.USER_ID != A.USER_ID;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
```

```
OPEN COMBINATIONS;
read_loop: LOOP
   SET EXIT1 = 0;
          SET F1_ACC = 0;
          SET F2_ACC = 0;
          SET F3_ACC = 0;
          SET TEMP = 0;


   FETCH COMBINATIONS INTO F1,F2,F3;
   IF done THEN
      LEAVE read_loop;
   END IF;
   SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F1_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F1 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
          SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F2_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F2 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
          SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F3_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F3 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;

          if (F1_ACC = 0 || F2_ACC = 0 || F3_ACC = 0) then
      set EXIT1 = -100;
          end if;

          if (EXIT1 = 0) then
      set TEMP = F1_ACC+F2_ACC+F3_ACC;
          end if;

          if (TOTAL < TEMP) then
      set TOTAL = TEMP;
          end if;
END LOOP;
CLOSE COMBINATIONS;

SET done = 0;
OPEN COMBINATIONS;
read_loop: LOOP
          SET EXIT1 = 0;
          SET F1_ACC = 0;
          SET F2_ACC = 0;
          SET F3_ACC = 0;
          SET TEMP = 0;

   FETCH COMBINATIONS INTO F1,F2,F3;
   IF done THEN
      LEAVE read_loop;
   END IF;
   SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F1_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F1 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
          SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F2_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F2 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
          SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F3_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F3 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;

          if (F1_ACC = 0 || F2_ACC = 0 || F3_ACC = 0) then
      set EXIT1 = -100;
          end if;

          if (EXIT1 = 0) then
      set TEMP = F1_ACC+F2_ACC+F3_ACC;
          end if;

          if (TOTAL = TEMP) then
      set F1_TOP3 =F1;
          end if;
END LOOP;
CLOSE COMBINATIONS;

SET done = 0;
OPEN COMBINATIONS;
read_loop: LOOP
          SET EXIT1 = 0;
          SET F1_ACC = 0;
          SET F2_ACC = 0;
          SET F3_ACC = 0;
          SET TEMP = 0;
```

```
  FETCH COMBINATIONS INTO F1,F2,F3;
  IF done THEN
    LEAVE read_loop;
  END IF;
  SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F1_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F1 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
      SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F2_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F2 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
      SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F3_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F3 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;

      if (F1_ACC = 0 || F2_ACC = 0 || F3_ACC = 0) then
    set EXIT1 = -100;
      end if;

      if (EXIT1 = 0) then
    set TEMP = F1_ACC+F2_ACC+F3_ACC;
      end if;

      if (TOTAL = TEMP) then
    set F2_TOP3 =F2;
      end if;
END LOOP;
CLOSE COMBINATIONS;

SET done = 0;
OPEN COMBINATIONS;
read_loop: LOOP
      SET EXIT1 = 0;
      SET F1_ACC = 0;
      SET F2_ACC = 0;
      SET F3_ACC = 0;
      SET TEMP = 0;
  FETCH COMBINATIONS INTO F1,F2,F3;
  IF done THEN
    LEAVE read_loop;
  END IF;
  SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F1_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F1 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
        SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F2_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F2 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;
        SELECT COUNT(COMMENTER_USER_ID) AS ACC INTO F3_ACC FROM COMMENTS WHERE
COMMENTER_USER_ID = F3 AND COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE
USER_ID = 10 OR USER_ID = F1 OR USER_ID=F2 OR USER_ID=F3) GROUP BY
COMMENTS.COMMENTER_USER_ID HAVING COUNT(COMMENTS.COMMENTER_USER_ID) >= 3;

        if (F1_ACC = 0 || F2_ACC = 0 || F3_ACC = 0) then
     set EXIT1 = -100;
        end if;

        if (EXIT1 = 0) then
     set TEMP = F1_ACC+F2_ACC+F3_ACC;
        end if;

        if (TOTAL = TEMP) then
     set F3_TOP3 =F3;
        end if;
END LOOP;
CLOSE COMBINATIONS;

SELECT USERS.USER_ID , USERS.NAME , F1ACC.ACC, TOTALCOMMENTS.ACC FROM USERS, (SELECT
COMMENTER_USER_ID,COUNT(COMMENTER_USER_ID) AS ACC FROM COMMENTS WHERE
COMMENTS.POST_ID NOT IN ( SELECT POST_ID FROM POSTS WHERE USER_ID = 10 OR USER_ID =
F1_TOP3 OR USER_ID=F2_TOP3 OR USER_ID=F3_TOP3) GROUP BY
COMMENTS.COMMENTER_USER_ID) F1ACC , ( SELECT
COMMENTER_USER_ID,COUNT(COMMENTER_USER_ID) AS ACC FROM COMMENTS GROUP BY
COMMENTER_USER_ID) TOTALCOMMENTS WHERE USERS.USER_ID IN ( F1_TOP3, F2_TOP3, F3_TOP3)
AND TOTALCOMMENTS.COMMENTER_USER_ID = USERS.USER_ID AND F1ACC.COMMENTER_USER_ID
= USERS.USER_ID ORDER BY F1ACC.ACC DESC;

END$$
DELIMITER ;

CALL TOP3ACC();
```