CS585 - Spring 2018 - HW2 Solution + Rubrics

Rubrics

0 Points If there are no explanations found in the submission.

Q1-5: **1 point** for each correct query

-1 for each query that doesn't work for any reason. *No Fractional points.*

Q6: **1 point** based on "students effort" and comparison with the provided solution. Run the query and check for yourself the expected output. Student should pick a ***right approach*** doing it, or not. Also a ***description*** **(like a pseudo-algorithm)** of what they have tried to do (even though not very successful in coding it) is very important.

-1 point to the overall grade if the name of the tables are not as mentioned in the HW description. No penalty if the difference is just due to upper/lowercase

-1 point If the students have not considered the UNIQUE ids for any of the queries (i.e. they have redundant outputs), deduct their point for that query.

Solution

Q1

```
SELECT DISTINCT COMMENTER_USER_ID AS ID,NAME
FROM   COMMENTS,USERS
WHERE  USER_ID = COMMENTER_USER_ID
       AND COMMENTER_USER_ID NOT IN (SELECT USER_ID
                                     FROM   POSTS)
       AND COMMENTER_USER_ID IN (SELECT COMMENTER_USER_ID
                                 FROM   COMMENTS
                                 WHERE  POST_ID = 5);
```

Q2

```
SELECT F.USER_ID
FROM   FRIENDSHIPS F,USERS U
WHERE  U.GENDER = 'F'
       AND F.USER_ID IN (SELECT F.FRIEND_ID
                         FROM   FRIENDSHIPS F
                         WHERE  F.USER_ID = 1)
       AND F.USER_ID IN (SELECT F.FRIEND_ID
                         FROM   FRIENDSHIPS F
                         WHERE  F.USER_ID = 2)
GROUP  BY F.USER_ID;
```

Q3

```
SELECT DISTINCT FR.USER_ID
-- if you want the # of friends, add this part: , CF.FRIEND_COUNT
FROM    (SELECT U.USER_ID
         FROM    USERS U,FRIENDSHIPS F
         WHERE   U.USER_ID = F.USER_ID
                 AND F.FRIEND_ID IN (SELECT USER_ID
                                     FROM    POSTS)) FR
        JOIN (SELECT U.USER_ID,Count(F.FRIEND_ID) AS FRIEND_COUNT
              FROM    USERS U,FRIENDSHIPS F
              WHERE   U.USER_ID = F.USER_ID
              GROUP   BY U.USER_ID) CF
          ON FR.USER_ID = CF.USER_ID
             AND CF.FRIEND_COUNT > 2;
```

Q4

```
SELECT FEMCOMM.USER_ID,FR.COUNT_FRIEND
FROM(SELECT U.USER_ID
     FROM    USERS U
     WHERE   U.DATE_OF_BIRTH > Date('1990-12-20')
          AND U.GENDER = 'F'
          AND U.USER_ID IN (SELECT COMMENTER_USER_ID
                            FROM    COMMENTS
                            WHERE   COMMENTS.POST_ID IN (SELECT
POST_ID
                                                         FROM    POSTS
P
                                                         WHERE
                                                    P.USER_ID = 10)))
        FEMCOMM
        JOIN (SELECT U.USER_ID,Count(F.FRIEND_ID) AS COUNT_FRIEND
              FROM    USERS U,FRIENDSHIPS F
              WHERE   U.USER_ID = F.USER_ID
              GROUP   BY U.USER_ID) FR
          ON FR.USER_ID = FEMCOMM.USER_ID;
```

Q5

```
SELECT COMMENTER_USER_ID
FROM    COMMENTS
WHERE   POST_ID = 7
        AND COMMENTER_USER_ID IN(SELECT F.FRIEND_ID
                                 FROM    FRIENDSHIPS F,POSTS P
```

```
                                    WHERE   F.USER_ID = P.USER_ID
                                            AND P.POST_ID = 7);
```

Q6

```
SELECT
F1 AS USER_ID, N1 AS NAME, ACC1 AS ACC, TOTAL_COMMENTS
FROM
(SELECT
AUGMENTED.F1, AUGMENTED.N1, AUGMENTED.ACC1
FROM
(SELECT
F1, F2, F3, N1, N2, N3, ACC1, ACC2, ACC3, ACC1 + ACC2 + ACC3
FROM
(SELECT
F1, F2, F3, N1, N2, N3, ACC1, ACC2, COUNT(COMMENT_ID) AS ACC3
FROM
(SELECT
F1, F2, F3, N1, N2, N3, ACC1, COUNT(COMMENT_ID) AS ACC2
FROM
(SELECT
F1, F2, F3, N1, N2, N3, COUNT(COMMENT_ID) AS ACC1
FROM
(SELECT
COL1.USER_ID AS F1, COL2.USER_ID AS F2, COL3.USER_ID AS F3, COL1.NAME AS N1,
COL2.NAME AS N2, COL3.NAME AS N3
FROM
(SELECT
USER_ID, NAME
FROM
USERS
WHERE
USER_ID IN (SELECT
USER_ID
FROM
FRIENDSHIPS
WHERE
FRIEND_ID = 20 AND GENDER ='F')) COL1
CROSS JOIN
(SELECT
USER_ID, NAME
FROM
USERS
WHERE
```

```
USER_ID IN (SELECT
USER_ID
FROM
FRIENDSHIPS
WHERE
FRIEND_ID = 20 AND GENDER = 'F')) COL2
CROSS JOIN
(SELECT
USER_ID, NAME
FROM
USERS
WHERE
USER_ID IN (SELECT
USER_ID
FROM
FRIENDSHIPS
WHERE
FRIEND_ID = 20 AND GENDER = 'F')) COL3
WHERE
COL1.USER_ID <> COL2.USER_ID AND COL1.USER_ID <> COL3.USER_ID AND
COL2.USER_ID <> COL3.USER_ID) TRIPLETS
, (SELECT
USER_ID AS POSTER, COMMENT_ID, COMMENTER_USER_ID AS COMMENTER
FROM
POSTS, COMMENTS
WHERE POSTS.POST_ID = COMMENTS.POST_ID AND POSTS.USER_ID <> 10)
POST_COMMENTS1
WHERE
F1 = COMMENTER AND POSTER <> F1 AND POSTER <> F2 AND POSTER <> F3
GROUP BY
F1 , F2 , F3) COUNT1
, (SELECT
USER_ID AS POSTER, COMMENT_ID, COMMENTER_USER_ID AS COMMENTER
FROM
POSTS, COMMENTS
WHERE
POSTS.POST_ID = COMMENTS.POST_ID AND POSTS.USER_ID <> 10)
POST_COMMENTS2
WHERE F2 = COMMENTER AND POSTER <> F1 AND POSTER <> F2 AND POSTER <> F3
GROUP BY
F1 , F2 , F3) COUNT2
, (SELECT
USER_ID AS POSTER, COMMENT_ID, COMMENTER_USER_ID AS COMMENTER
```

```
FROM
POSTS, COMMENTS
WHERE
POSTS.POST_ID = COMMENTS.POST_ID AND POSTS.USER_ID <> 10)
POST_COMMENTS3
WHERE
F3 = COMMENTER AND POSTER <> F1 AND POSTER <> F2 AND POSTER <> F3
GROUP BY
F1, F2, F3) COUNT3
GROUP BY
F1, F2, F3) AUGMENTED
, (SELECT F1, F2, F3
FROM
(SELECT
F1, F2, F3, ACC1, ACC2, ACC3, ACC1 + ACC2 + ACC3 AS TOTAL_ACC
FROM
(SELECT
F1, F2, F3, ACC1, ACC2, COUNT(COMMENT_ID) AS ACC3
FROM
(SELECT
F1, F2, F3, ACC1, COUNT(COMMENT_ID) AS ACC2
FROM
(SELECT
F1, F2, F3, COUNT(COMMENT_ID) AS ACC1
FROM
(SELECT
COL1.USER_ID AS F1, COL2.USER_ID AS F2, COL3.USER_ID AS F3
FROM
(SELECT
USER_ID
FROM
FRIENDSHIPS
WHERE
USER_ID IN (SELECT
USER_ID
FROM
USERS
WHERE
FRIEND_ID = 20 AND GENDER = 'F')) COL1
CROSS JOIN (SELECT
USER_ID
FROM
FRIENDSHIPS
```

```
WHERE
USER_ID IN (SELECT
USER_ID
FROM
USERS
WHERE
FRIEND_ID = 20 AND GENDER = 'F')) COL2
CROSS JOIN (SELECT
USER_ID
FROM
FRIENDSHIPS
WHERE
USER_ID IN (SELECT
USER_ID
FROM
USERS
WHERE
FRIEND_ID = 20 AND GENDER = 'F')) COL3
WHERE
COL1.USER_ID <> COL2.USER_ID AND COL1.USER_ID <> COL3.USER_ID AND
COL2.USER_ID <> COL3.USER_ID) TRIPLETS
, (SELECT
USER_ID AS POSTER, COMMENT_ID, COMMENTER_USER_ID AS COMMENTER
FROM
POSTS, COMMENTS
WHERE
POSTS.POST_ID = COMMENTS.POST_ID AND POSTS.USER_ID <> 10)
POST_COMMENTS1
WHERE
F1 = COMMENTER AND POSTER <> F1 AND POSTER <> F2 AND POSTER <> F3
GROUP BY
F1 , F2 , F3) COUNT1
,(SELECT
USER_ID AS POSTER, COMMENT_ID, COMMENTER_USER_ID AS COMMENTER
FROM
POSTS, COMMENTS
WHERE
POSTS.POST_ID = COMMENTS.POST_ID AND POSTS.USER_ID <> 10)
POST_COMMENTS2
WHERE
F2 = COMMENTER AND POSTER <> F1 AND POSTER <> F2 AND POSTER <> F3
GROUP BY
F1 , F2 , F3) COUNT2
```

```
,(SELECT
USER_ID AS POSTER, COMMENT_ID, COMMENTER_USER_ID AS COMMENTER
FROM
POSTS, COMMENTS
WHERE
POSTS.POST_ID = COMMENTS.POST_ID AND POSTS.USER_ID <> 10)
POST_COMMENTS3
WHERE
F3 = COMMENTER AND POSTER <> F1 AND POSTER <> F2 AND POSTER <> F3
GROUP BY
F1 , F2 , F3) COUNT3
GROUP BY
F1 , F2 , F3) TOTAL_COUNT
WHERE
ACC1 >= 3 AND ACC2 >= 3 AND ACC3 >= 3
ORDER BY
TOTAL_ACC DESC LIMIT 1) COUNTED
WHERE
(AUGMENTED.F1 = COUNTED.F1 AND AUGMENTED.F2 = COUNTED.F2 AND
AUGMENTED.F3 = COUNTED.F3)
OR
(AUGMENTED.F1 = COUNTED.F2 AND AUGMENTED.F2 = COUNTED.F1 AND
AUGMENTED.F3 = COUNTED.F3)
OR
(AUGMENTED.F1 = COUNTED.F3 AND AUGMENTED.F2 = COUNTED.F2 AND
AUGMENTED.F3 = COUNTED.F1)) RANKED
,(SELECT
COMMENTER_USER_ID, COUNT(COMMENT_ID) AS TOTAL_COMMENTS
FROM
COMMENTS
GROUP BY
COMMENTER_USER_ID) TOTAL_COUNT
WHERE
RANKED.F1 = TOTAL_COUNT.COMMENTER_USER_ID
ORDER BY
ACC DESC;
```