-- Query [1]

SELECT \* FROM "USER"

WHERE rname IS NOT NULL

ORDER BY rname, name;

-- Query [2]

SELECT UNIQUE AID, CATEGORY

FROM FMENTION, FCATEGORY

WHERE FMENTION.FID = FCATEGORY.FID;

-- Query [3]

SELECT UNIQUE USER.ID AS UID, ARTICLE.ID AS AID

FROM USER, ARTICLE

EXCEPT ALL

SELECT UID, AID

FROM COMMENT;

-- Query [4]

SELECT UNIQUE pp.AID, i.ID

FROM FMENTION pp

CROSS JOIN

(SELECT UNIQUE f.ID AS pid, g.ID AS ID

FROM FILM f, FILM g

WHERE f.ID <> g.ID

AND NOT EXISTS(

SELECT \*

FROM FCATEGORY b

WHERE f.ID = b.FID AND

b.CATEGORY NOT IN (

SELECT p.CATEGORY

FROM FCATEGORY p

WHERE p.CATEGORY IN

(SELECT l.CATEGORY

FROM FCATEGORY l

WHERE l.FID = g.ID)

)

)

AND NOT EXISTS(

SELECT \*

FROM FCATEGORY b

WHERE g.ID = b.FID AND

b.CATEGORY NOT IN (

SELECT p.CATEGORY

FROM FCATEGORY p

WHERE p.CATEGORY IN

(SELECT l.CATEGORY

FROM FCATEGORY l

WHERE l.FID = f.ID)

)

)

) i

WHERE i.pid = pp.FID

;

-- Query [5]

SELECT a.AID, a.TITLE,

(SELECT RNAME FROM USER WHERE a.AID = ID) AS NAME,

(SELECT COUNT(\*) FROM COMMENT c WHERE c.ON\_ID IS NULL AND c.AID = a.ID) AS TCOUNT,

(SELECT COUNT(\*) FROM COMMENT c WHERE c.AID = a.ID) AS RCOUNT

FROM ARTICLE a

ORDER BY a.RTIME DESC;

-- Query [6]

WITH table2 (UID, CATEGORY, SCORE) AS

(SELECT table1.UID, table1.CATEGORY, COUNT(table1.CATEGORY) AS SCORE

FROM (SELECT UNIQUE c.UID, f.AID, f2.CATEGORY

FROM COMMENT c

INNER JOIN FMENTION f ON c.AID = f.AID

INNER JOIN FCATEGORY f2 ON f2.FID = f.FID

ORDER BY c.UID) AS table1

GROUP BY table1.CATEGORY, table1.UID)

SELECT t.UID, t.CATEGORY, t.SCORE

FROM table2 t

WHERE NOT EXISTS(

SELECT t2.UID, t2.CATEGORY, t2.SCORE

FROM table2 t2

WHERE t2.UID = t.UID AND t2.SCORE > t.SCORE

)

;

-- Query [7]

WITH PERMUTATIONS (CATEGORY, CATEGORY2) AS

(SELECT UNIQUE f.CATEGORY AS CATEGORY, f1.CATEGORY AS CATEGORY2

FROM FCATEGORY f, FCATEGORY f1

WHERE f.CATEGORY <> f1.CATEGORY

ORDER BY f.CATEGORY)

,

NUMBERPER (CATEGORY, CATEGORY2, COUNTED) AS

(SELECT pp.CATEGORY, pp.CATEGORY2, COUNT(pp.catFID) AS COUNTED

FROM (SELECT UNIQUE p.CATEGORY, p.CATEGORY2, cat.FID AS catFID

FROM PERMUTATIONS p, FCATEGORY cat

WHERE p.CATEGORY IN(

SELECT cat2.CATEGORY

FROM FCATEGORY cat2

WHERE cat2.FID = cat.FID

)

AND p.CATEGORY2 IN(

SELECT cat2.CATEGORY

FROM FCATEGORY cat2

WHERE cat2.FID = cat.FID

) ) pp

GROUP BY pp.CATEGORY, pp.CATEGORY2

ORDER BY pp.CATEGORY, pp.CATEGORY2)

,

COMMON(CATEGORY, CATEGORY2) AS

(SELECT b.CATEGORY, b.CATEGORY2

FROM NUMBERPER b

WHERE b.CATEGORY2 IN(

SELECT c.CATEGORY2

FROM NUMBERPER c

WHERE c.CATEGORY = b.CATEGORY

AND NOT EXISTS(

SELECT cc.CATEGORY2

FROM NUMBERPER cc

WHERE cc.CATEGORY = c.CATEGORY

AND cc.COUNTED > c.COUNTED

)

)

)

SELECT DISTINCT c2.CATEGORY AS CATEGORY, c1.CATEGORY AS FCATEGORY, c1.FID AS FID

FROM FCATEGORY c1, FCATEGORY c2

WHERE c1.FID <> c2.FID

AND c2.CATEGORY NOT IN(

SELECT pp.CATEGORY

FROM FCATEGORY pp

WHERE pp.FID = c1.FID

)

AND EXISTS(

SELECT UNIQUE c.CATEGORY, c.CATEGORY2

FROM COMMON c

WHERE c.CATEGORY = c2.CATEGORY AND c.CATEGORY2 = c1.CATEGORY

)

ORDER BY c1.FID

;

--Query [8]

WITH MENTIONED(UID ,ID, ON\_ID) AS

(SELECT c.UID, c.ID, c.ON\_ID

FROM COMMENT c

ORDER BY ON\_ID)

,

REACTIONS(ORIGINAL, PARENTID, CHILDID) AS

(SELECT DISTINCT c.UID AS ORIGINAL, c.ID AS PARENTID, d.ID AS CHILDID

FROM MENTIONED c, MENTIONED d

WHERE c.ID = d.ON\_ID)

,

NUMREACTIONS (ORIGINAL, PARENTID, NUMREAC) AS

(SELECT DISTINCT c.ORIGINAL AS ORIGINAL, c.PARENTID AS PARENTID, COUNT (c.CHILDID)

FROM REACTIONS c

GROUP BY c.PARENTID, c.ORIGINAL)

,

NUMBEROFCOMMENTS (TOP) AS (SELECT DISTINCT SUM(n.NUMREAC) AS TOP

FROM NUMREACTIONS n)

,

NUMBEROFCOMMENTSTOTAL (TOP) AS (SELECT COUNT(c.ID) AS TOP

FROM COMMENT c)

,

AVERAGEREPLIES (AMOUNT) AS (SELECT c.TOP \* 1.0 / c1.TOP \* 2 AS AMOUNT

FROM NUMBEROFCOMMENTS c, NUMBEROFCOMMENTSTOTAL c1)

,

NUMOFCOMMENTSREPLIES(ORIGINAL, NUMCOMMENTS, NUMREACTIONS) AS

(SELECT pp.ORIGINAL, pp.NUMCOMMENTS, ppp.NUMREACTIONS

FROM (SELECT r.UID AS ORIGINAL, COUNT(r.ID) AS NUMCOMMENTS FROM COMMENT r

GROUP BY r.UID) pp

INNER JOIN

(SELECT r.ORIGINAL, SUM(r.NUMREAC) AS NUMREACTIONS FROM NUMREACTIONS r

GROUP BY ORIGINAL) ppp

ON pp.ORIGINAL = ppp.ORIGINAL

)

,

CLICKS(ORIGINAL, AVERAGECLICKS, AMOUNT) AS (SELECT ORIGINAL, NUMREACTIONS \* 1.0 / NUMCOMMENTS AS AVERAGECLICKS, AMOUNT

FROM NUMOFCOMMENTSREPLIES, AVERAGEREPLIES

ORDER BY ORIGINAL)

SELECT c.ORIGINAL AS UID

FROM CLICKS c

WHERE c.AVERAGECLICKS > c.AMOUNT

;

-- Query [9]

WITH MAIN(AUTHOR, FID, RTIME, SCORE) AS

(SELECT DISTINCT a.AID AS AUTHOR, f.FID, a.RTIME, f.SCORE

FROM ARTICLE a

RIGHT JOIN

FMENTION f

ON f.AID = a.ID

ORDER BY a.AID, f.FID, a.RTIME)

,

PREVIOUS (AUTHOR, FID, RTIME, SCORE, AUTHOR2, RTIME2, PREVSCORE) AS

(SELECT m.AUTHOR, m.FID, m.RTIME, m.SCORE, mm.AUTHOR AS AUTHOR2, mm.RTIME AS RTIME2, mm.SCORE AS PREVSCORE

FROM MAIN m

LEFT JOIN (SELECT q.AUTHOR, q.FID, q.RTIME, q.SCORE FROM MAIN q) mm

ON m.FID = mm.FID

AND m.AUTHOR = mm.AUTHOR

AND m.RTIME > mm.RTIME

AND NOT EXISTS (

SELECT pop.RTIME

FROM MAIN pop

WHERE pop.FID = m.FID

AND pop.AUTHOR = m.AUTHOR

AND pop.RTIME < m.RTIME

AND pop.RTIME > mm.RTIME

)

)

,

CLEAROUT (AUTHOR, FID, RTIME, SCORE, AUTHOR2, RTIME2, PREVSCORE) AS

(SELECT \* FROM PREVIOUS b

WHERE PREVSCORE IS NOT NULL

)

,

INCREASEGROUPS (FID, COUNTED) AS

(SELECT FID, COUNT(FID) AS NUMARTICLES

FROM CLEAROUT

WHERE PREVSCORE < SCORE

GROUP BY FID

)

,

TOTAL (FID, COUNTED) AS

(SELECT FID, COUNT(FID) AS NUMARTICLES

FROM CLEAROUT

GROUP BY FID

)

,

DISPLAY(FID, N, M) AS (SELECT t.FID, t.COUNTED AS N, inc.COUNTED AS M

FROM TOTAL t

LEFT JOIN

INCREASEGROUPS inc

ON t.FID = inc.FID

ORDER BY FID)

,

NOTNULL(FID, VAL) AS (SELECT FID, M - N

FROM DISPLAY

WHERE M IS NOT NULL

)

,

NULLED(FID, VAL) AS (SELECT FID, 0 - N

FROM DISPLAY

WHERE M IS NULL

)

,

END (FID, TS) AS (SELECT \*

FROM NOTNULL

UNION

SELECT \*

FROM NULLED

ORDER BY FID)

SELECT \* FROM END

;

-- Query [10] IDK