**COMP 533 HW2**

**Mo Tang mt60 S01276578**

**1.**

(a)

SELECT COUNT(\*) From(

SELECT name, email, phone FROM central\_contacts AS c1

UNION ALL

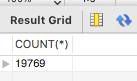
SELECT name, email, phone FROM result\_contacts AS c2

UNION ALL

SELECT name, email, phone FROM central\_contacts AS c3)

As total;

Count: 19769



(b)

SELECT COUNT(\*) From(

SELECT name, email, phone FROM central\_contacts AS c1

UNION

SELECT name, email, phone FROM result\_contacts AS c2

UNION

SELECT name, email, phone FROM central\_contacts AS c3)

As total;

Count: 9296



(c)

In current database, the same person may have different name. Like “MD” with “M.D.” We could put the name in a more standard way, like every name can only have one abbreviation, and divide the whole name to Job Title, First Name, Last Name.

**2.**

(a) The value has every condition.

SELECT SUM(p.points) AS highScore FROM scorePoints AS p WHERE p.points>0;

Value: 89



(b)

SELECT MAX(s2.score) FROM

(SELECT nct\_id, SUM(sp.points) AS score

FROM

(SELECT s.nct\_id, t.term

FROM studies AS s

INNER JOIN conditions AS c ON c.nct\_id = s.nct\_id INNER JOIN scoreTerms AS t ON t.name = c.name GROUP BY s.nct\_id, t.term)

AS t

INNER JOIN scorePoints AS sp ON sp.term = t.term GROUP BY t.nct\_id)

AS s2;

Score value : 23



(c)

SELECT COUNT(\*)

FROM(

SELECT nct\_id, SUM(sp.points) AS score

FROM(

SELECT s.nct\_id, t.term

FROM studies AS s

INNER JOIN conditions AS co ON co.nct\_id = s.nct\_id

INNER JOIN scoreTerms AS t ON t.name = co.name AND t.term = "Neurodegenerative disorders"

GROUP BY s.nct\_id, t.term)

AS t

INNER JOIN scorePoints AS sp ON sp.term = t.term GROUP BY t.nct\_id) study\_score WHERE score = 6;



(d) studies: 11018

SELECT COUNT(\*)

FROM studies AS s

WHERE s.nct\_id NOT IN

(SELECT nct\_id FROM

(SELECT DISTINCT s.nct\_id

FROM studies AS s

INNER JOIN

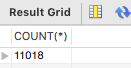
(SELECT \*

FROM conditions AS co

WHERE co.name IN (SELECT name FROM scoreTerms))

AS

r ON r.nct\_id = s.nct\_id) AS risk);



(e)

SELECT round(avg(num),2) AS NUMBER1 FROM (

SELECT nct\_id, COUNT(i) AS num

FROM

(SELECT s.nct\_id, co.id AS i

FROM studies AS s

INNER JOIN conditions AS co ON co.nct\_id = s.nct\_id

INNER JOIN scoreTerms AS st ON st.name = co.name

GROUP BY s.nct\_id, co.id)

AS ns

GROUP BY nct\_id) AS non;



SELECT ROUND(AVG(num),2) AS NUMBER2 FROM (

SELECT nct\_id, count(term) AS num

FROM

(SELECT nct\_id, term

FROM

(SELECT s.nct\_id, st.term

FROM studies AS s

INNER JOIN conditions AS co ON co.nct\_id = s.nct\_id INNER JOIN scoreTerms AS st ON st.name = co.name GROUP BY s.nct\_id, st.term)

AS st

WHERE st.nct\_id IN ( SELECT nct\_id FROM

(SELECT nct\_id, SUM(sp.points) AS score

FROM

(SELECT s.nct\_id, st.term

FROM studies AS s

INNER JOIN conditions AS co ON co.nct\_id = s.nct\_id INNER JOIN scoreTerms AS st ON st.name = co.name GROUP BY s.nct\_id, st.term)

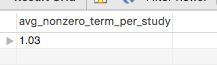
AS st

INNER JOIN scorePoints AS sp ON sp.term = st.term GROUP BY st.nct\_id)

AS sco WHERE score !=0))

AS nn

GROUP BY nct\_id) AS non;



**3.**

The current tables have : “term-points”, we just need add a new table:”term-newPoints”. The identical data structure will make all the operation the same.

**4.**

The terms with negative points influence the final score.

**5.**

CREATE VIEW study AS

SELECT DISTINCT st.nct\_id, co.name AS na FROM studies AS st

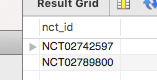
INNER JOIN conditions AS co ON st.nct\_id = co.nct\_id GROUP BY st.nct\_id, co.name;

SELECT DISTINCT nct\_id

FROM study AS ca WHERE NOT EXISTS

(SELECT cb.na FROM study AS cb WHERE cb.nct\_id = "NCT02789800" AND

cb.na NOT IN (SELECT na FROM study AS cc WHERE cc.nct\_id = ca.nct\_id) );



**6.**

(a)

0.0204; NCT02742597-NCt02595866 , NCT02742597-NCT03002311

CREATE VIEW jindex AS

SELECT j.id1, j.id2, ROUND(( j.common/(j.val\_a+j.val\_b - j.common)),4) AS ja FROM

(SELECT ic.id1, ic.id2, ic.inter\_cnt AS common, gc1.cnt AS val\_a, gc2.cnt AS val\_b FROM

(SELECT id1, id2, COUNT(name) AS inter\_cnt FROM

(SELECT c1.id AS id1, c2.id AS id2, c1.name AS name

FROM (SELECT c.nct\_id AS id, c.name AS name FROM conditions AS c) AS c1

CROSS JOIN (select c.nct\_id AS id, c.name AS name FROM conditions AS c) AS c2 ON c1.id < c2.id AND c1.name = c2.name)

GROUP BY id1, id2)

AS ic

INNER JOIN

(SELECT c.id AS id, COUNT(c.name) AS cnt FROM (SELECT c.nct\_id AS id, c.name AS name FROM conditions AS c) AS c

GROUP BY c.id)

AS gc1 ON gc1.id = ic.id1

INNER JOIN

(SELECT c.id AS id, COUNT(c.name) AS cnt FROM (SELECT c.nct\_id AS id, c.name AS name FROM conditions AS c) AS c

GROUP BY c.id)

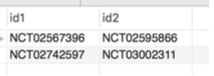
AS gc2 ON gc2.id = ic.id2)

AS j;

SELECT min(j.ja) from jindex as res;



SELECT j.id1, j.id2 from jindex as j where j.jacard = 0.0204;



(b)

0.3653

SELECT ROUND( cnt\_1/total,4) AS percentage FROM

(SELECT COUNT(ja) AS cnt\_1 FROM jindex

WHERE ja = 1.0000)

,(SELECT COUNT(ja) AS total FROM jindex);

