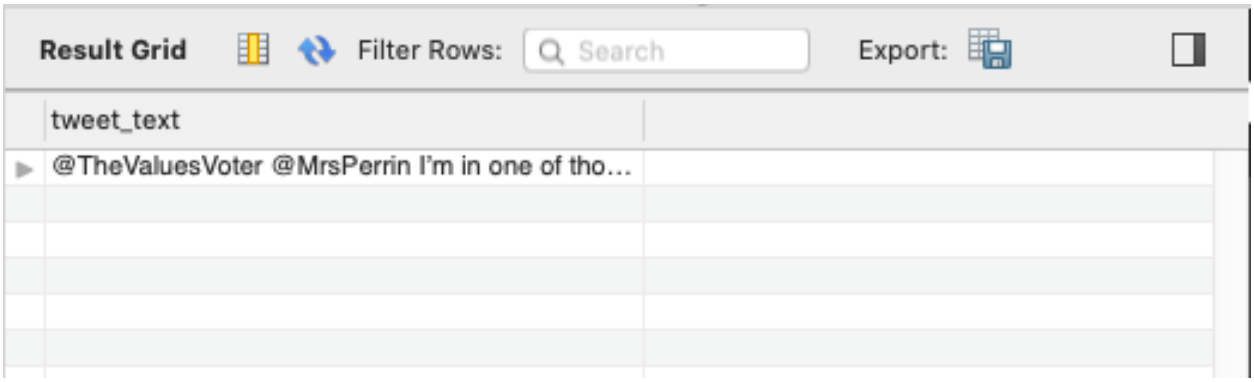
Structured Query Language

1. Find the text of all tweets that were posted by the tweeter with the handle ‘patgotweet’.

a) SQL Query:

```
1 •  SELECT tweet_text
2    FROM Tweet A, Tweeter B
3    WHERE A.tweeter_id = B.tweeter_id AND B.handle= 'patgotweet'
```

b) Result: (1 Row)

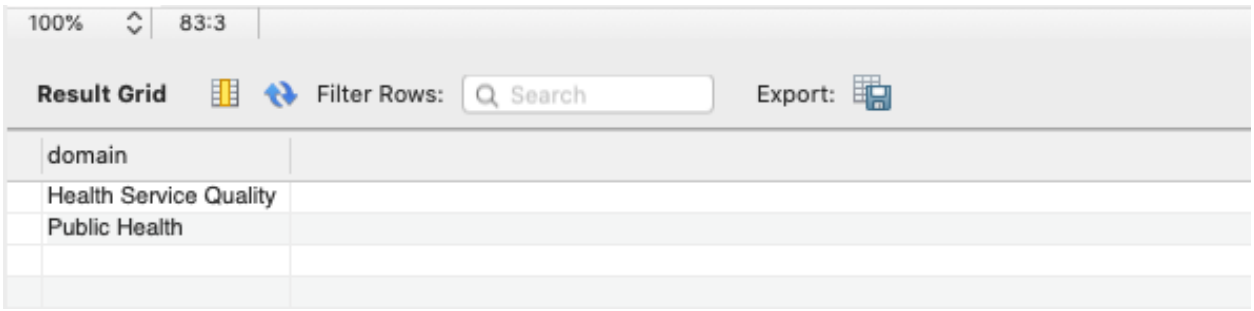


2. List the **distinct** domains of expertise for checkers who have verified tweets that have the hashtag “*COVID19*”. (**Note: The hashtag value is all in capital letters.)**

a) SQL Query:

```
1 •  SELECT DISTINCT E.domain
2    FROM Expertise E, Verification V, Hashtags H
3    WHERE H.hashtag = 'COVID19' AND V.user_id = E.user_id AND V.tweet_id = H.Tweet_id;
4
```

b) Result: (2 Rows)



3. List the handles of Tweeters who have posted a tweet that has been verified by a Checker who started as a checker after the date “2020-01-31 03:41:49”.

a) SQL Query:

```
1 •  SELECT DISTINCT Tr.handle
2    FROM Tweeter Tr, Tweet T, Verification V, Checker C
3    WHERE C.checker_since > '2020-01-31 03:41:49'
4    AND C.user_id=V.user_id AND V.tweet_id=T.tweet_id AND T.tweeter_id = Tr.tweeter_id;
```

b) Result: (5 Rows)

100% 84:4

Result Grid Filter Rows: Search Export:

handle
NecessaryPaper
oceanviewmom
SandyInCalif
theblack_abyss
mptrottier

4. For verified tweets that contain the hashtag "COVID19", find the associated evidence URLs, verification comments, and checkers' first and last names **(Again: “COVID19” is in all caps.)**

a) SQL Query:

```
1 •  SELECT DISTINCT Ev.URL, V.comment, U.name_first, U.name_last
2    FROM User U, Verification V, Evidence Ev, Hashtags Ht, VerifiedUsing VU
3    WHERE Ht.hashtag='COVID19' AND Ht.tweet_id = V.tweet_id
4    AND V.ver_id = VU.ver_id AND EV.ev_id = VU.ev_id AND U.user_id=V.User_id
5
```

b) Result (12 rows):

100% 59:2

Result Grid Filter Rows: Search Export:

URL	comment	name_first	name_last
http://states-covid-numbers.org	Masks works! Check the CDC	Jonathan	Howard
http://florida-covid19.gov	Masks works! Check the CDC	Jonathan	Howard
https://cdc.gov	Masks works! Check the CDC	Jonathan	Howard
http://florida-covid19.gov	Masks works! Check the CDC	Antonio	Olson
https://cdc.gov	Masks works! Check the CDC	Antonio	Olson
http://states-covid-numbers.org	Masks works! Check the CDC	Antonio	Olson
http://states-covid-numbers.org	Masks works! Check the CDC	Gina	Miranda
http://florida-covid19.gov	Masks works! Check the CDC	Gina	Miranda
https://cdc.gov	Masks works! Check the CDC	Gina	Miranda
https://cdc.gov	Masks works! Check the CDC	Courtney	White
http://mask-works.info	Masks works! Check the CDC	Courtney	White
http://covid-is-not-hoax.net	Masks works! Check the CDC	Courtney	White

5. Find the user IDs, first names, and last names of checkers that have **all** the domains of expertise from the user with ID = 68. (Note: Your answer will include the "ID = 68" checker as well, of course.)

a) SQL Query:

```
1  SELECT U.user_id, U.name_first, U.name_last
2  FROM User U
3  WHERE NOT EXISTS (SELECT E.domain
4                    FROM Expertise E
5                    WHERE E.user_id = 68 AND
6                      NOT EXISTS(
7                          SELECT E2.domain
8                          FROM Expertise E2
9                          WHERE E2.domain = E.domain
10                         AND E2.user_id = U.user_id
11                     )
12                 )
```

b) Result: (3 Rows)

100% 1:1

Result Grid Filter Rows: Search Export:

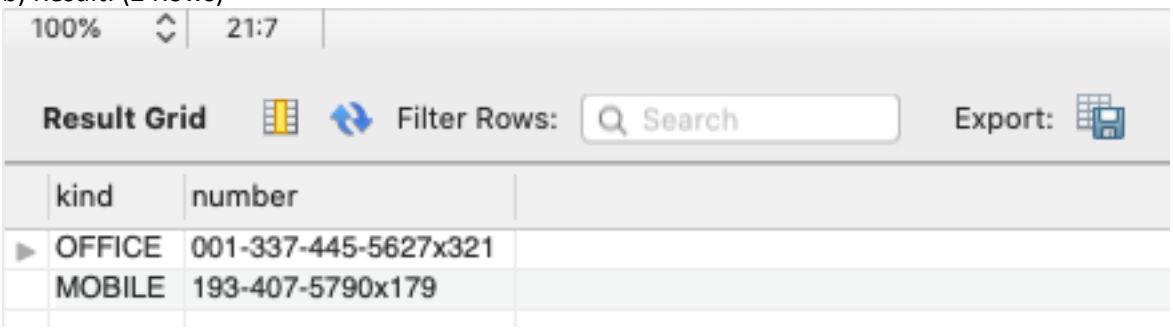
user_id	name_first	name_last
40	Lauren	Rhodes
68	Darren	Ortiz
87	Anthony	Monroe

6. List the phone numbers of checkers who have verified the tweet with the id "1321211561046933514" ***and*** who are experts in "Infectious Diseases" (Note the use of the word "and" instead of "or" from the previous assignment!)

a) SQL Query:

```
SELECT DISTINCT P.kind, P.number
FROM Phone P
WHERE P.user_id IN (SELECT E.user_id
                    FROM Expertise E
                    WHERE E.domain='Infectious Diseases')
  AND P.user_id IN (SELECT V.user_id
                    FROM Verification V
                    WHERE V.tweet_id='1321211561046933514')
```

b) Result: (2 Rows)



100% 21:7

Result Grid Filter Rows: Search Export:

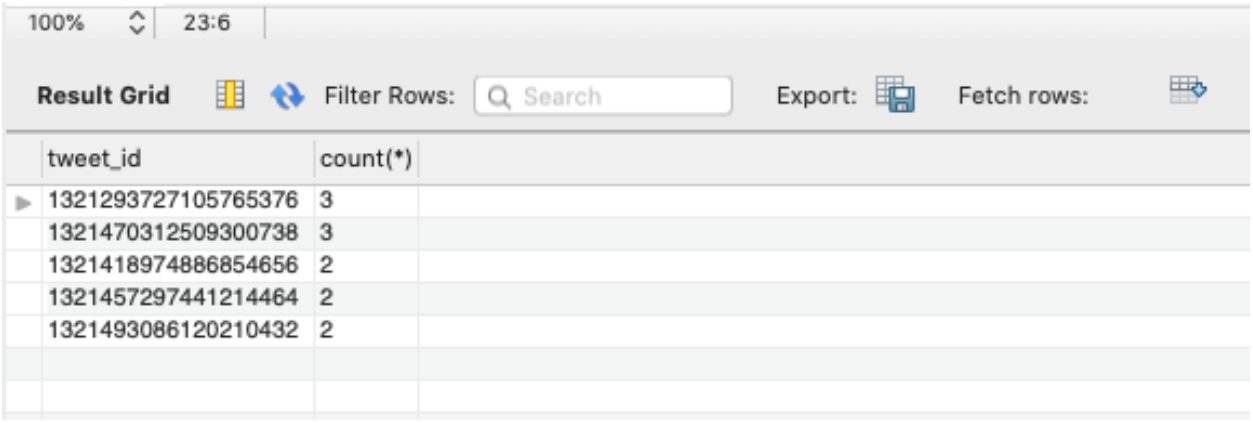
kind	number
OFFICE	001-337-445-5627x321
MOBILE	193-407-5790x179

7. Find tweet ids and the number of replies for each tweet that has one or more replies. List only the top five tweets that have the highest number of replies.

a) SQL Query:

```
1  SELECT T1.tweet_id, count(*)
2  FROM Tweet T1, Tweet T2
3  WHERE T2.replied_to_tweet=T1.tweet_id
4  GROUP BY T1.tweet_id
5  Having count(*)>0
6  ORDER BY count(*) DESC
7  LIMIT 5
```

b) Result: (5 Rows)



100% 23:6

Result Grid Filter Rows: Search Export: Fetch rows:

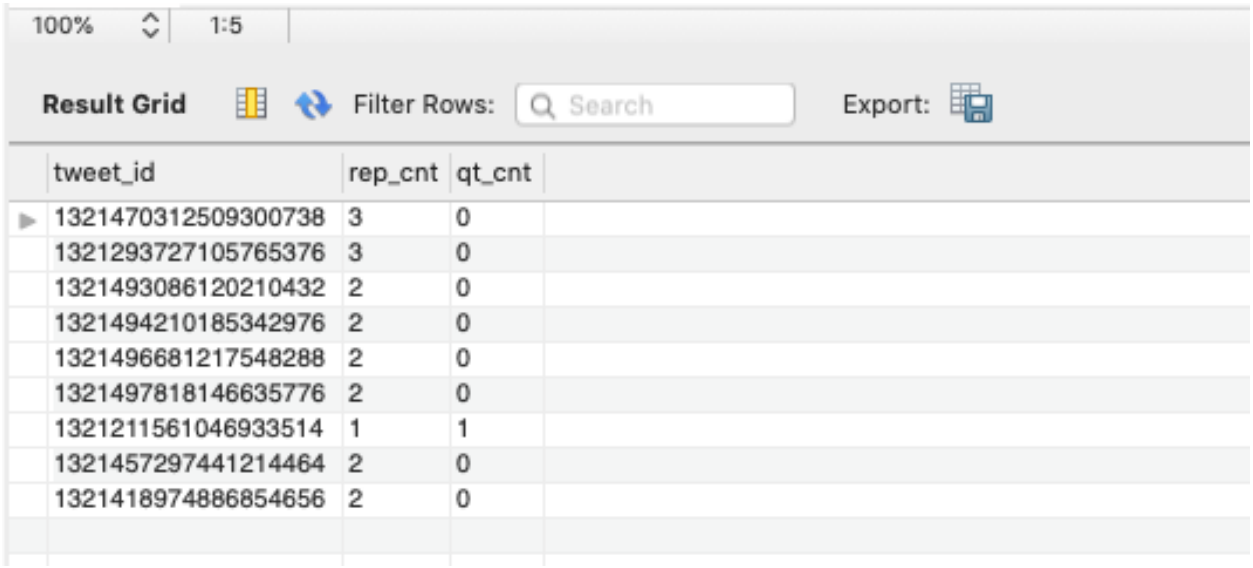
tweet_id	count(*)
1321293727105765376	3
1321470312509300738	3
1321418974886854656	2
1321457297441214464	2
1321493086120210432	2

8. For tweets that have two or more reactions (replies and/or quotes), print their tweet id along with their number of replies and number of quotes. (Note that for such tweets, the sum of replies and quotes should be 2 or more). Order the result by the number of reactions in largest-first order.

a) SQL Query:

```sql
SELECT T1.tweet_id, (SELECT count(*) FROM Tweet T2 WHERE T2.replied_to_tweet=T1.tweet_id) AS rep_cnt,
(SELECT count(*) FROM Tweet T3 WHERE T3.quoted_tweet=T1.tweet_id) AS qt_cnt
FROM Tweet T1
WHERE (SELECT count(*) FROM Tweet T2 WHERE T2.replied_to_tweet=T1.tweet_id) +
(SELECT count(*) FROM Tweet T3 WHERE T3.quoted_tweet=T1.tweet_id) >=2
ORDER BY rep_cnt+qt_cnt DESC
```

b) Result (9 rows):



tweet_id	rep_cnt	qt_cnt
1321470312509300738	3	0
1321293727105765376	3	0
1321493086120210432	2	0
1321494210185342976	2	0
1321496681217548288	2	0
1321497818146635776	2	0
1321211561046933514	1	1
1321457297441214464	2	0
1321418974886854656	2	0