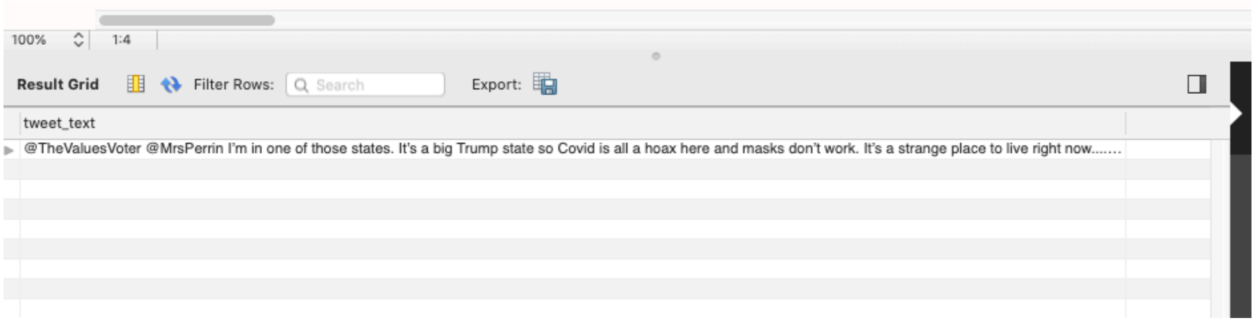
Last Name: yiting **First Name:** wang **Student ID:** 13853462

1. [10 pts] Find the text of all tweets that were posted by the tweeter with the handle ‘patgotweet’.

a) [7 pts] SQL Query:

```
select Tweet.tweet_text

From Tweet,Tweeter

WHERE Tweet.tweeter_id=Tweeter.tweeter_id AND Tweeter.handle='patgotweet'
```

b) [3 pts] Result: (1 Row)



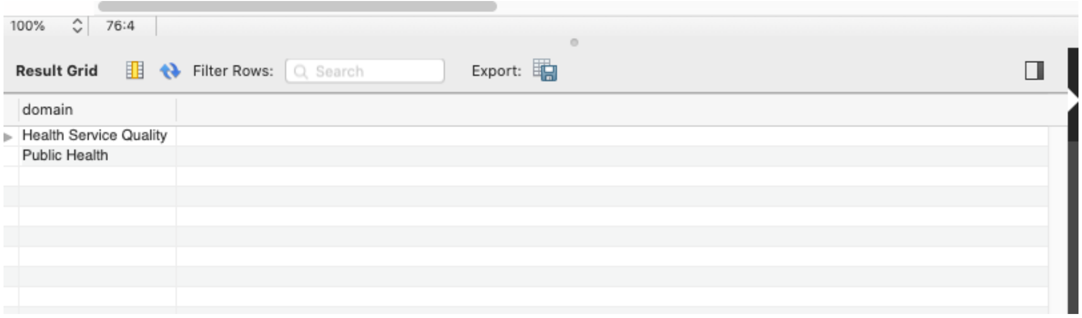
tweet_text
@TheValuesVoter @MrsPerrin I’m in one of those states. It’s a big Trump state so Covid is all a hoax here and masks don’t work. It’s a strange place to live right now.......

2. [10 pts] 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) [7 pts] SQL Query:

```
select Distinct Expertise.domain

From Expertise,Hashtags,Verification

WHERE Expertise.user_id= Verification.user_id

        And Verification.tweet_id=Hashtags.tweet_id AND Hashtags.hashtag='COVID19'
```

b) [3 pts] Result: (2 Rows)



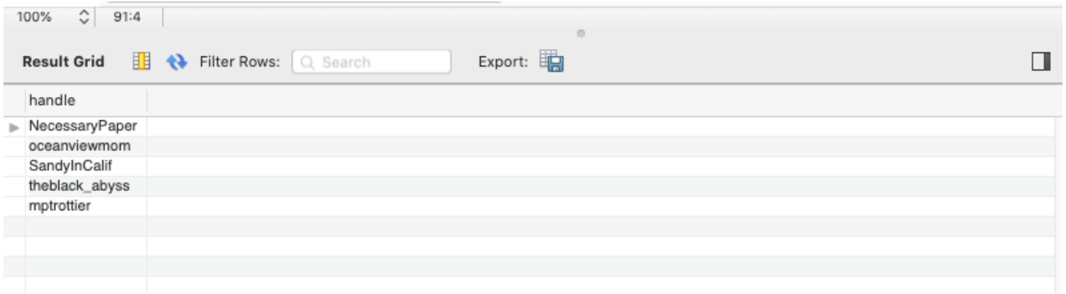
```
domain
Health Service Quality
Public Health
```

3. [10 pts] 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) [7 pts] SQL Query:

```
select distinct Tweeter.handle

From Tweeter,Tweet,Verification,Checker

WHERE Tweeter.tweeter_id=Tweet.tweeter_id AND Tweet.tweet_id=Verification.tweet_id

        AND Verification.user_id=Checker.user_id AND Checker.checker_since>='2020-01-31 03:41:49'
```

b) [3 pts] Result: (5 Rows)



```
handle
NecessaryPaper
oceanviewmom
SandyInCalif
theblack_abyss
mptrottier
```

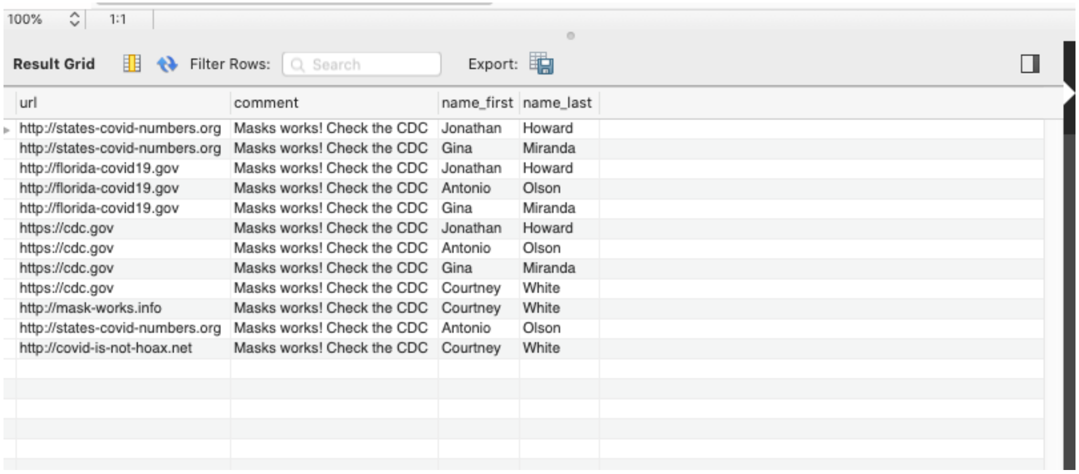
4. [15 pts] For verified tweets that contain the hashtag "COVID19", find the associated evidence URLs, verification comments, and checkers' first and last names **(Again: “COVID19” is in all caps.)**

a) [12 pts] SQL Query:

```
select DISTINCT Evidence.url,Verification.comment, User.name_first, User.name_last

From Evidence,VerifiedUsing,Verification,User,Tweet,Hashtags
```

```
WHERE Evidence.ev_id=VerifiedUsing.ev_id AND VerifiedUsing.ver_id=Verification.ver_id
        AND Verification.tweet_id=Hashtags.tweet_id AND Hashtags.hashtag='COVID19'
        AND Verification.user_id=User.user_id
```

b) [3 pts] Result (12 rows):



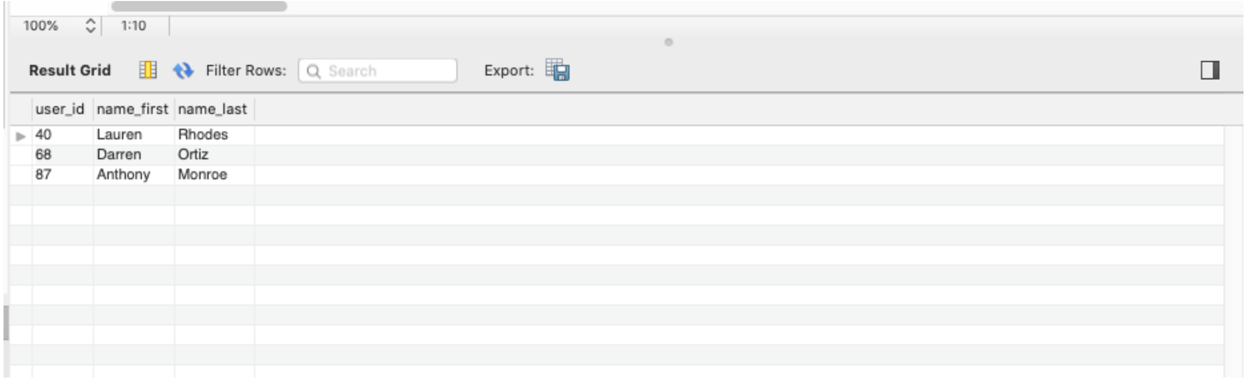
url	comment	name_first	name_last
http://states-covid-numbers.org	Masks works! Check the CDC	Jonathan	Howard
http://states-covid-numbers.org	Masks works! Check the CDC	Gina	Miranda
http://florida-covid19.gov	Masks works! Check the CDC	Jonathan	Howard
http://florida-covid19.gov	Masks works! Check the CDC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

5. [15 pts] 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) [12 pts] SQL Query:

```
SELECT User.user_id, User.name_first, User.name_last
FROM User
WHERE NOT EXISTS
(SELECT * FROM Expertise E2
WHERE E2.user_id = 68 AND NOT EXISTS (SELECT *
FROM Expertise E1
WHERE E1.user_id = User.user_id AND E2.domain = E1.domain))
```

b) [3 pts] Result: (3 Rows)



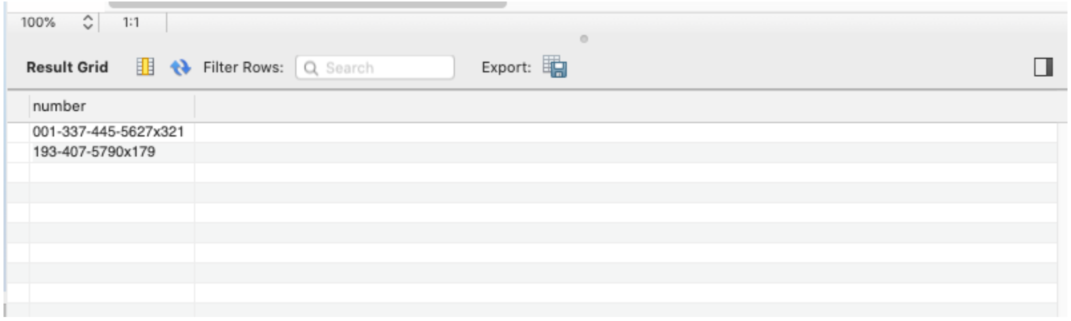
user_id	name_first	name_last
40	Lauren	Rhodes
68	Darren	Ortiz
87	Anthony	Monroe

6. [10 pts] 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) [7 pts] SQL Query:

```
select DISTINCT Phone.number

From Phone,Checker,Tweet,Expertise,Verification

WHERE Phone.user_id=Checker.user_id

        AND Checker.user_id=Expertise.user_id AND Expertise.domain='Infectious Diseases'

        AND Checker.user_id in (select Checker.user_id

                                        From Checker,Verification

                WHERE Checker.user_id=Verification.user_id AND
Verification.tweet_id="1321211561046933514"

                                        )
```

b) [3 pts] Result: (2 Rows)



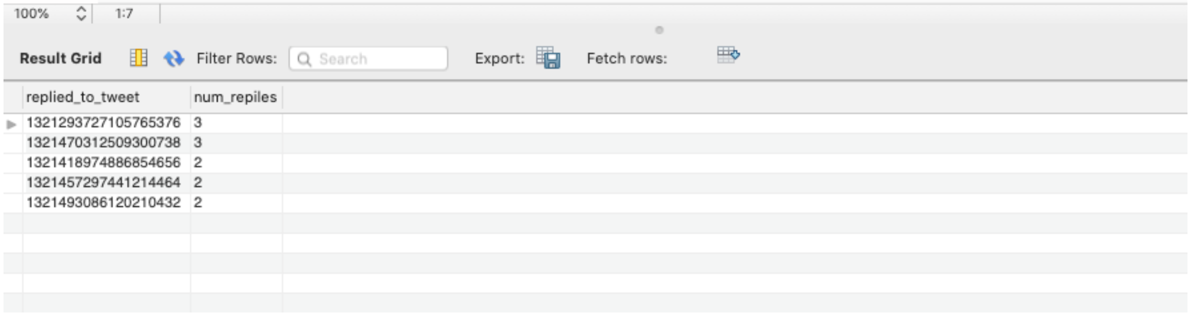
number
001-337-445-5627x321
193-407-5790x179

7. [15 pts] 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) [12 pts] SQL Query:

```
SELECT Tweet.replied_to_tweet, COUNT(Tweet.replied_to_tweet)

 FROM Tweet

GROUP BY Tweet.replied_to_tweet

HAVING COUNT(Tweet.replied_to_tweet) >= 1

ORDER BY COUNT(Tweet.replied_to_tweet) DESC

LIMIT 5
```

b) [3 pts] Result: (5 Rows)



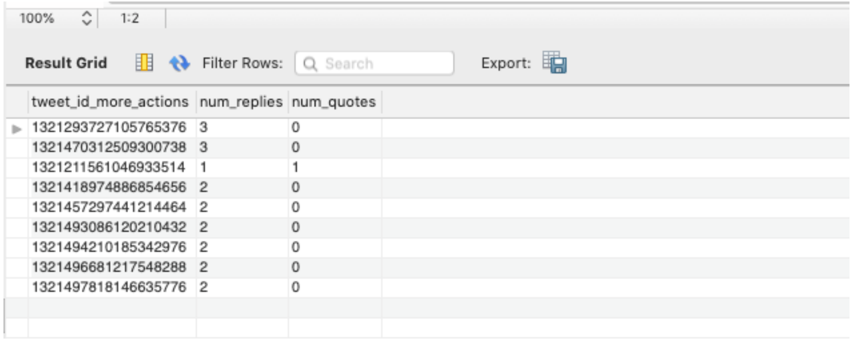
replied_to_tweet	num_repiles
1321293727105765376	3
1321470312509300738	3
1321418974886854656	2
1321457297441214464	2
1321493086120210432	2

8. [15 pts] 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) [12 pts] SQL Query:

```
SELECT T2.tweet_id_needed, COUNT(T1.replied_to_tweet) , COUNT(T1.quoted_tweet)

FROM ((SELECT quoted_tweet as tweet_id_needed

FROM Tweet
GROUP BY quoted_tweet)

UNION

(SELECT replied_to_tweet as tweet_id_needed

FROM Tweet

GROUP BY replied_to_tweet)

) T2, Tweet T1

WHERE T1.replied_to_tweet = T2.tweet_id_needed OR T1.quoted_tweet = T2.tweet_id_needed

GROUP BY T2.tweet_id_needed

HAVING COUNT(T1.replied_to_tweet)+COUNT(T1.quoted_tweet) >= 2

ORDER BY COUNT(T1.replied_to_tweet)+COUNT(T1.quoted_tweet) DESC;
```

b) [3 pts] Result (9 rows):



tweet_id_more_actions	num_replies	num_quotes
1321293727105765376	3	0
1321470312509300738	3	0
1321211561046933514	1	1
1321418974886854656	2	0
1321457297441214464	2	0
1321493086120210432	2	0
1321494210185342976	2	0
1321496681217548288	2	0
1321497818146635776	2	0