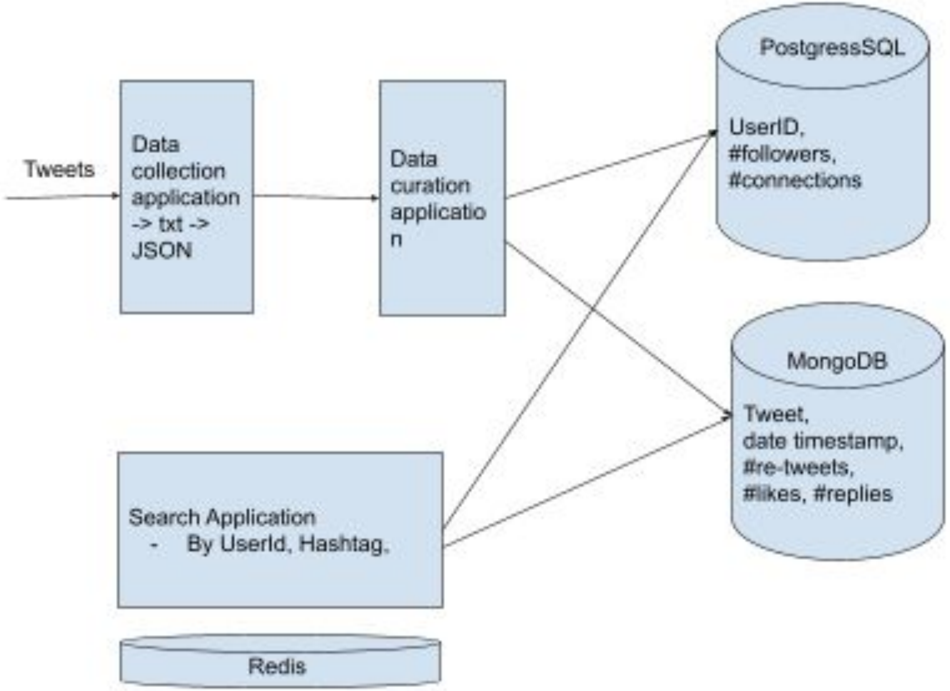
Tweet Search Application

Isha Raju and Shreeya Rajanarayanan

Data Collection

- Originally gather 30,000 tweets just using the hashtag Covid19
  - After removing duplicate retweets, we only had around 7,500 tweets
  - For the sake of complexity, we decided to gather more
- Regathered a total of 50,000 tweets using the hashtags Covid19, crypto
  - After parsing through tweets in one go, we had 15,475 entries for user information and 10,947 entries for unique tweets
  - Both topics are prevalent in the news these days and make an interesting topic
  - We only gathered tweets in english
  - Gathered tweets on 4/23 stored in a json file
  - Uninterrupted

Design


Tweets
Data collection application -> txt -> JSON
Data curation applicatio n
PostgressSQL
UserID, #followers, #connections
MongoDB
Tweet, date timestamp, #re-tweets, #likes, #replies
Search Application
- By UserId, Hashtag,
Redis

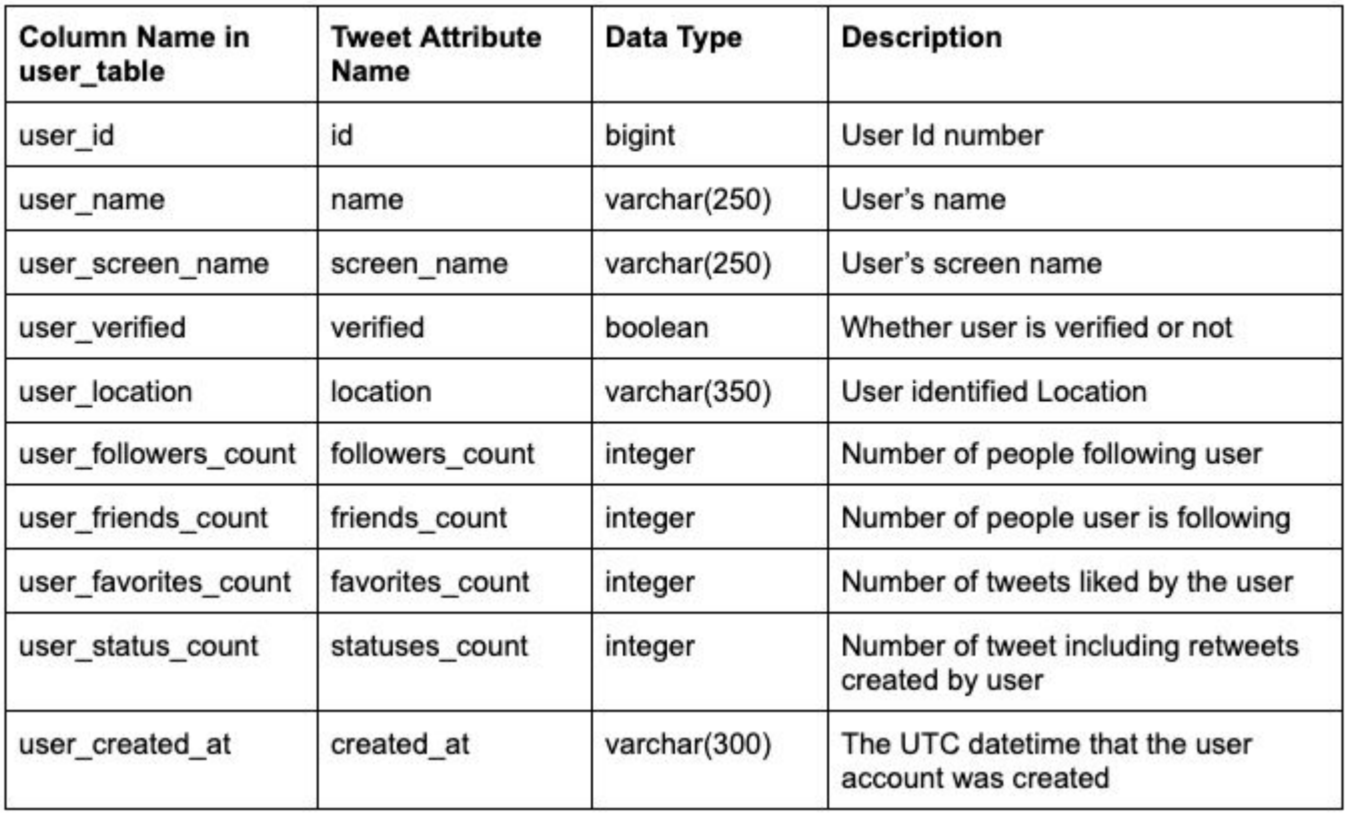
Database selection

- We chose to use PostgreSQL for our relational database since it was easy to install and use on our systems
- We chose MongoDB because the tweets are unstructured and there are a varying number of nested json objects in our data. MongoDB allowed us to store these JSON style documents. There is also a lot of documentation available for MongoDB, which makes using it more accessible.
- We chose Redis to cache instead of creating a dictionary because using Redis was easier.

Data Storage Process

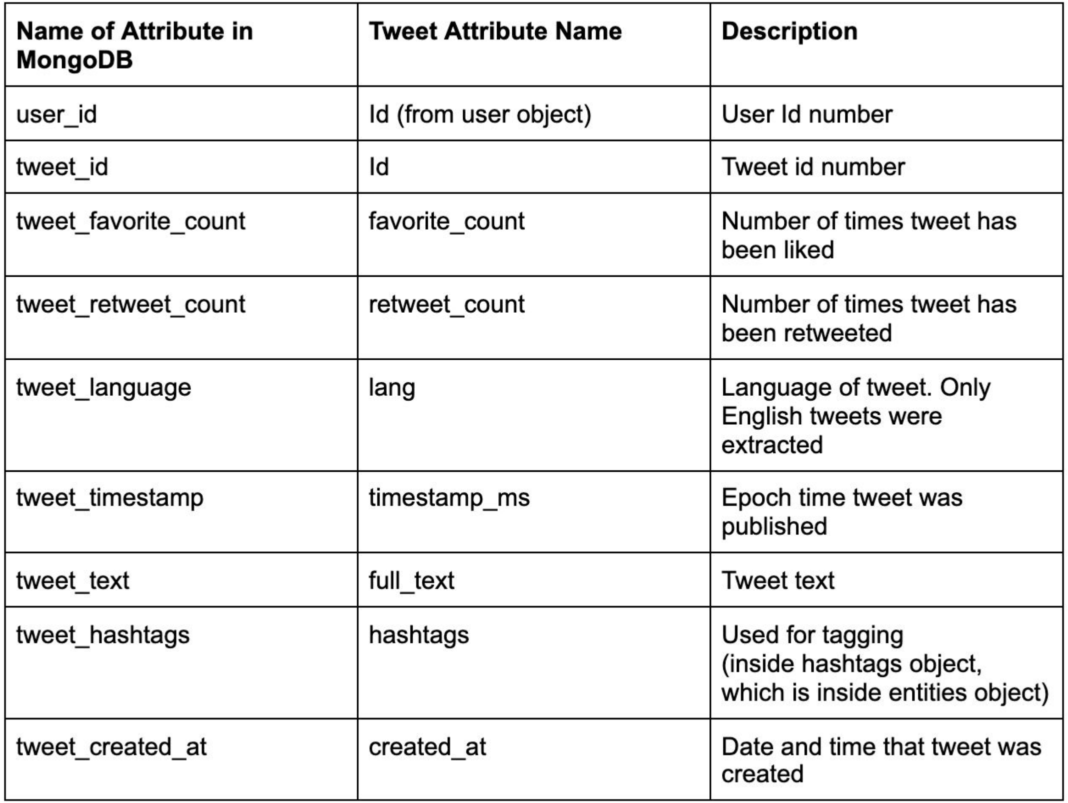
- Iterated through each tweet and considered 2 cases
    - Case 1 (Regular Tweet)
        - All **user** information gathered from the user json object nested within each general tweet object.
        - Most tweet information would be gathered from the greater/general tweet object
            - **Hashtags** were taken from the entities nested object
            - If there was an extended tweet the **tweet text** was taken from the extended_tweet object. if there was no extended tweet the text was taken from the general tweet object
    - Case 2 (Retweet)
        - **User** info gathered from the user object within the retweeted_status object within the general object
        - Most tweet information would be gathered from the retweeted_status object
            - **Hashtags** were taken from the entities nested object within the retweeted_status object
            - **Tweet text** taken from extended_tweet object within the retweeted_status object if it's an extended tweet or it's taken from the retweeted_status object
            -
    - After gathering all the info, user info was saved in PostgeSQL and unique tweet info was saved in MongoDB

User Data Storage: Postgres



Column Name in user_table	Tweet Attribute Name	Data Type	Description
user_id	id	bigint	User Id number
user_name	name	varchar(250)	User’s name
user_screen_name	screen_name	varchar(250)	User’s screen name
user_verified	verified	boolean	Whether user is verified or not
user_location	location	varchar(350)	User identified Location
user_followers_count	followers_count	integer	Number of people following user
user_friends_count	friends_count	integer	Number of people user is following
user_favorites_count	favorites_count	integer	Number of tweets liked by the user
user_status_count	statuses_count	integer	Number of tweet including retweets created by user
user_created_at	created_at	varchar(300)	The UTC datetime that the user account was created

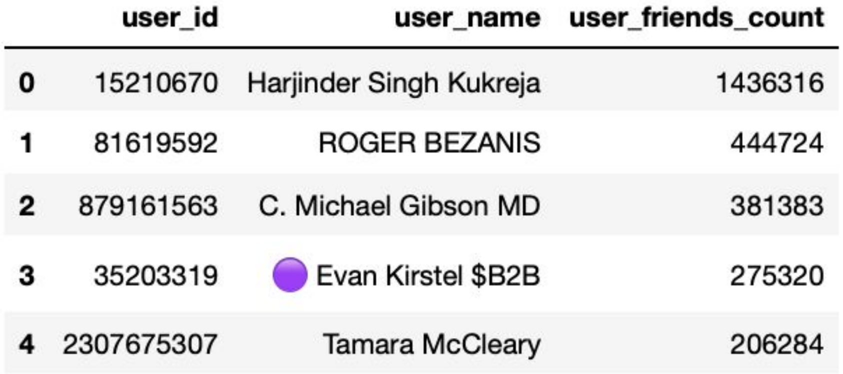
Tweet Data Storage: MongoDB



Name of Attribute in MongoDB	Tweet Attribute Name	Description
user_id	Id (from user object)	User Id number
tweet_id	Id	Tweet id number
tweet_favorite_count	favorite_count	Number of times tweet has been liked
tweet_retweet_count	retweet_count	Number of times tweet has been retweeted
tweet_language	lang	Language of tweet. Only English tweets were extracted
tweet_timestamp	timestamp_ms	Epoch time tweet was published
tweet_text	full_text	Tweet text
tweet_hashtags	hashtags	Used for tagging (inside hashtags object, which is inside entities object)
tweet_created_at	created_at	Date and time that tweet was created

Querying PostgreSQL Database

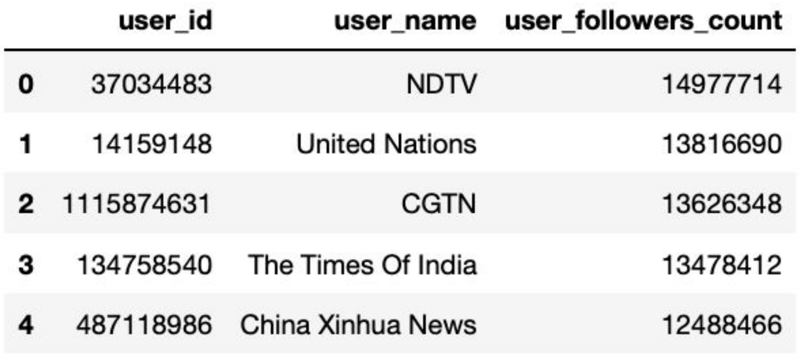
- Using SQL, we're able to query the user information database
- The following are some results for sorting users based on their friend count



	user_id	user_name	user_friends_count
0	15210670	Harjinder Singh Kukreja	1436316
1	81619592	ROGER BEZANIS	444724
2	879161563	C. Michael Gibson MD	381383
3	35203319	🟣 Evan Kirstel $B2B	275320
4	2307675307	Tamara McCleary	206284

Querying PostgreSQL Database

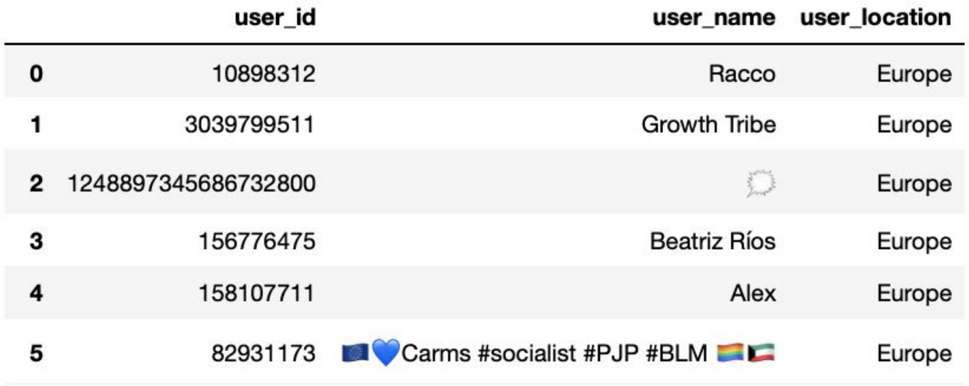
- The following are some results when we pull information about the number of followers a user has.



	user_id	user_name	user_followers_count
0	37034483	NDTV	14977714
1	14159148	United Nations	13816690
2	1115874631	CGTN	13626348
3	134758540	The Times Of India	13478412
4	487118986	China Xinhua News	12488466

Querying PostgreSQL Database

- The following are some results when we pull user information for users based in Europe



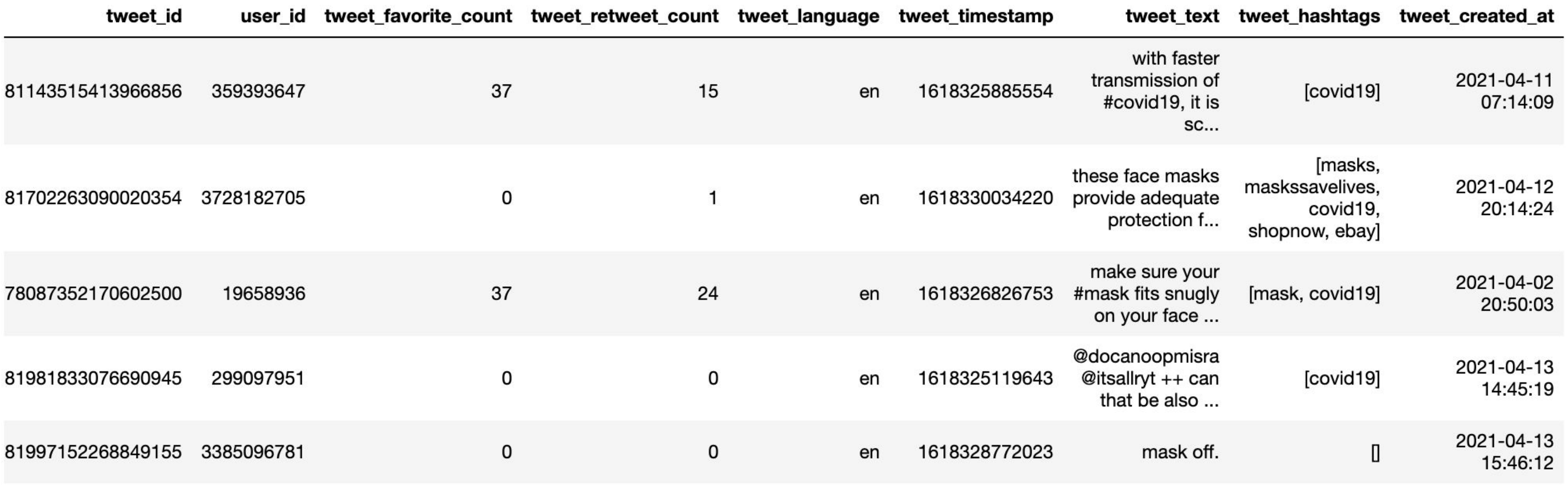
	user_id	user_name	user_location
0	10898312	Racco	Europe
1	3039799511	Growth Tribe	Europe
2	1248897345686732800	🗯	Europe
3	156776475	Beatriz Ríos	Europe
4	158107711	Alex	Europe
5	82931173	🇪🇺💙Carms #socialist #PJP #BLM 🏳️‍🌈🇰🇼	Europe

Search Application

- Added indexes to our tweet database
    - Indexed on tweet_id, tweet_text, tweet_hashtags, tweet_created_at
- Implemented cache using Redis
- Search application first checks if the query is stored in the redis cache. If they are, the results are taken from the cache. If the query was not cached, then the search application looks in MongoDB

Search Application Results - Keyword Search

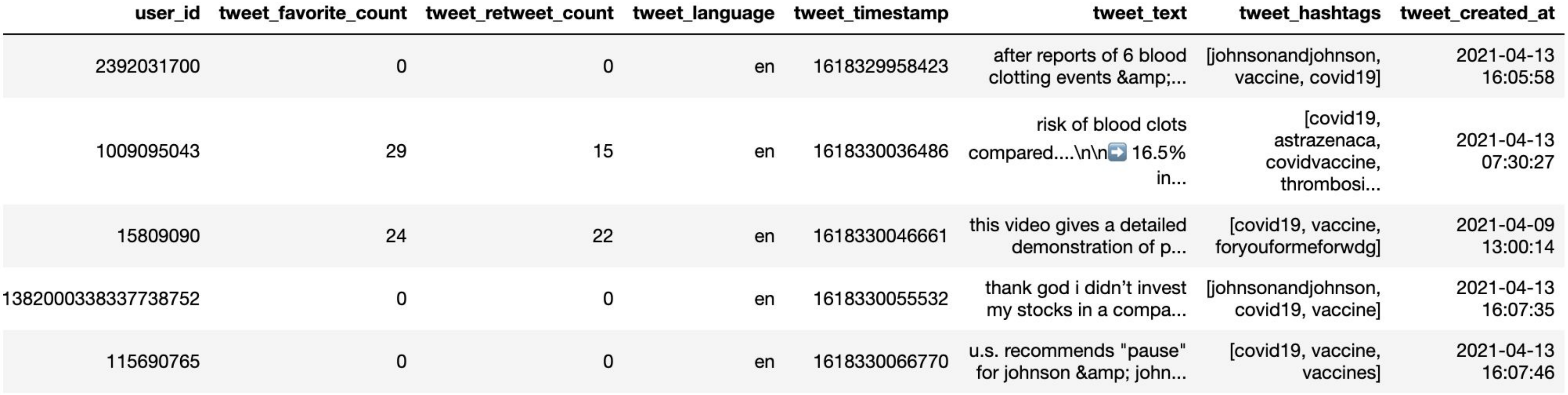
- Caching helped reduce search time. When searching by the key word “mask”, the search time is 0.327 seconds before the results were stored in the cache and 0.013 seconds after the results were stored in the cache



tweet_id	user_id	tweet_favorite_count	tweet_retweet_count	tweet_language	tweet_timestamp	tweet_text	tweet_hashtags	tweet_created_at
811435154139668 56	359393647	37	15	en	1618325885554	with faster transmission of #covid19, it is sc...	[covid19]	2021-04-11 07:14:09
81702263090020354	3728182705	0	1	en	1618330034220	these face masks provide adequate protection f...	[masks, maskssavelives, covid19, shopnow, ebay]	2021-04-12 20:14:24
78087352170602500	19658936	37	24	en	1618326826753	make sure your #mask fits snugly on your face ...	[mask, covid19]	2021-04-02 20:50:03
81981833076690945	299097951	0	0	en	1618325119643	@docanoopmisra @itsallryt ++ can that be also ...	[covid19]	2021-04-13 14:45:19
81997152268849155	3385096781	0	0	en	1618328772023	mask off.	[]	2021-04-13 15:46:12

Search Application Results - Hashtag

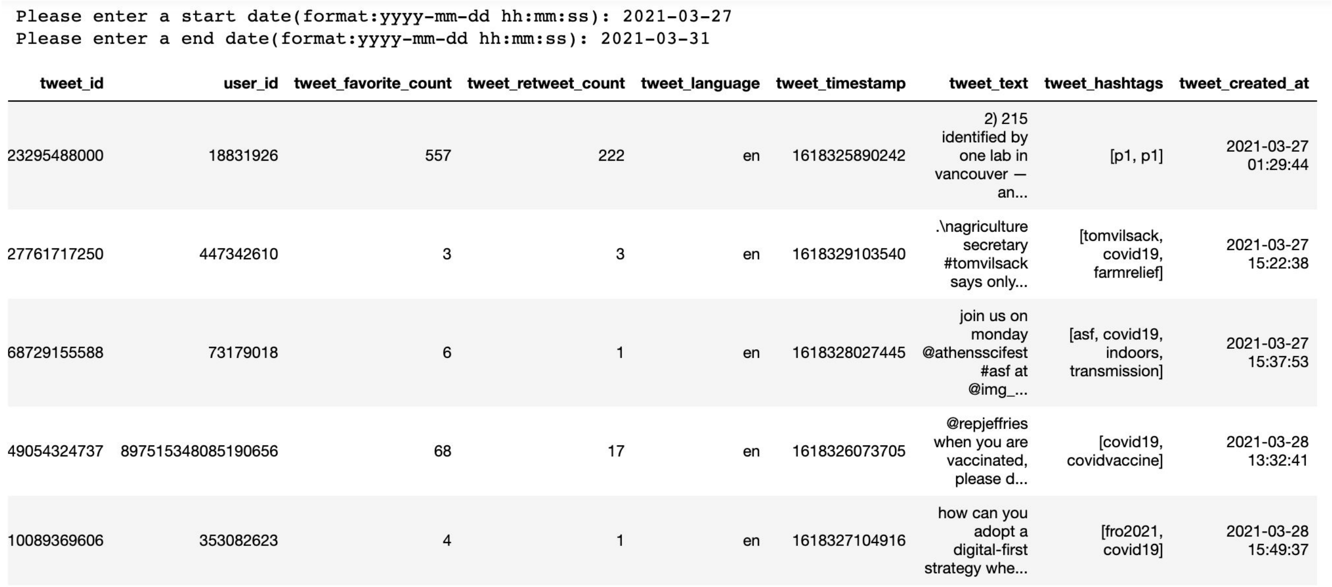
- When searching by the hashtag #vaccine, the search time is 0.018 seconds before the results were stored in the cache and 0.002 seconds after the results were stored in the cache



user_id	tweet_favorite_count	tweet_retweet_count	tweet_language	tweet_timestamp	tweet_text	tweet_hashtags	tweet_created_at
2392031700	0	0	en	1618329958423	after reports of 6 blood clotting events &amp;...	[johnsonandjohnson, vaccine, covid19]	2021-04-13 16:05:58
1009095043	29	15	en	1618330036486	risk of blood clots compared....\n\n➡️ 16.5% in...	[covid19, astrazenaca, covidvaccine, thrombosi...	2021-04-13 07:30:27
15809090	24	22	en	1618330046661	this video gives a detailed demonstration of p...	[covid19, vaccine, foryouformeforwdg]	2021-04-09 13:00:14
1382000338337738752	0	0	en	1618330055532	thank god i didn’t invest my stocks in a compa...	[johnsonandjohnson, covid19, vaccine]	2021-04-13 16:07:35
115690765	0	0	en	1618330066770	u.s. recommends "pause" for johnson &amp; john...	[covid19, vaccine, vaccines]	2021-04-13 16:07:46

Search Application Results - Search by date

- We’re also able to search the database and pull tweets created between a certain timeframe



```
Please enter a start date(format:yyyy-mm-dd hh:mm:ss): 2021-03-27
Please enter a end date(format:yyyy-mm-dd hh:mm:ss): 2021-03-31
```

tweet_id	user_id	tweet_favorite_count	tweet_retweet_count	tweet_language	tweet_timestamp	tweet_text	tweet_hashtags	tweet_created_at
23295488000	18831926	557	222	en	1618325890242	2) 215 identified by one lab in vancouver — an...	[p1, p1]	2021-03-27 01:29:44
27761717250	447342610	3	3	en	1618329103540	.\nagriculture secretary #tomvilsack says only...	[tomvilsack, covid19, farmrelief]	2021-03-27 15:22:38
68729155588	73179018	6	1	en	1618328027445	join us on monday @athensscifest #asf at @img_...	[asf, covid19, indoors, transmission]	2021-03-27 15:37:53
49054324737	897515348085190656	68	17	en	1618326073705	@repjeffries when you are vaccinated, please d...	[covid19, covidvaccine]	2021-03-28 13:32:41
10089369606	353082623	4	1	en	1618327104916	how can you adopt a digital-first strategy whe...	[fro2021, covid19]	2021-03-28 15:49:37

Final thoughts

- Surprisingly, only keeping unique information really had an impact on the size of our database and helped to streamline the design.
- We learned a lot about working with MongoDB, Postgres and Redis. This is our first exposure to these topics and they will be very useful for us in the future.