COMS 4121
Homework 1
Yiwen Zhang
yz3310

Step2: Protocol Buffers

1. Find the number of deleted messages in the dataset. –1554

   (code under next question)

2. Find the number of tweets that are replies to another tweet.--2531

```
>>> from twitter_pb2 import *
>>> tw = Tweets()
>>> f = open('twitter.pb', 'rb')
>>> tw.ParseFromString(f.read())
2531526
>>> f.close()
>>> deletenumber = 0
>>> replynumber = 0
>>> for t in tw.tweets:
...     insert = t.insert
...     if(t.is_delete==True):
...         deletenumber += 1
...     if insert.HasField('reply_to'):
...         replynumber += 1
...
>>> print("Number of deleted messages:",deletenumber)
('Number of deleted messages:', 1554)
>>> print("Number of reply messages:",replynumber)
('Number of reply messages:', 2531)
>>>
```

3. Find the five user IDs (field name: uid) that have tweeted the most.

rank	id	# of being tweeted
1	1269521828	5
2	392695315	4
3	424808364	3
4	1706901902	3
5	1471774728	2

```
>>> from collections import Counter
>>> tweets_uid=list()
>>> for k in tw.tweets:
...     ins=k.insert
...     if ins.uid !=long(0):
...             tweets_uid.append(ins.uid)
...
>>> count=Counter(tweets_uid)
>>> most_five=count.most_common(5)
>>> most_five
[(1269521828L, 5), (392695315L, 4), (424808364L, 3), (1706901902L, 3), (1471774728L, 2)]
>>>
```

Step3: SQL

1. Find the number of deleted messages in the dataset--1554
2. Find the number of tweets that are replies to another tweet--2531
3. Find the five user IDs (field name: uid) that have tweeted the most—

rank	id	# of being tweeted
1	1269521828	5
2	392695315	4
3	424808364	3
4	1706901902	3
5	23991910	2

```
sqlite> SELECT count(*) FROM tweets WHERE is_delete = 1;
1554
sqlite> SELECT count(*) FROM tweets WHERE reply_to != 0;
2531
sqlite> SELECT uid FROM tweets WHERE is_delete = 0 GROUP BY uid ORDER BY count(*) DESC LIMIT 5;
1269521828
392695315
424808364
1706901902
23991910
```

Step 4: MongoDB

1. Find the number of deleted messages in the dataset.--1554
2. Find the number of tweets that are replies to another tweet.--2531
3. Find the five user IDs (field name: `uid`) that have tweeted the most.

rank	id	# of being tweeted
1	1269521828	5
2	392695315	4
3	424808364	3
4	1706901902	3
5	578015986	2

```
> db.tweets.count({delete: {$exists: true}})
1554
> db.tweets.count({in_reply_to_status_id: {$exists: true, $ne: null}})
2531
> db.tweets.aggregate([
...           { $match: {delete: {$exists: false}}},
...           { $group: { _id: "$user.id", number: { $sum: 1 }}},
...           { $sort: { number: -1 }},
...           { $limit : 5 }
...         ])
{ "_id" : 1269521828, "number" : 5 }
{ "_id" : 392695315, "number" : 4 }
{ "_id" : 424808364, "number" : 3 }
{ "_id" : 1706901902, "number" : 3 }
{ "_id" : 578015986, "number" : 2 }
```

Reflection:

1. Read the schema and protocol buffer definition files. What are the main differences between the two? Are there any similarities?

   Differences: Protocol buffer are using programming language to perform. SQL schema use easy command to perform result in relational database management system.

   Similarities: They share same data structure and same types of data categories.

2. Describe one question that would be easier to answer with protocol buffers than via a SQL query.

   If we are asked to count the users who have the most friends, and the database is multi-nested query, protocol buffer would be better. In this situation, SQL need more computation time, because SQL needs to combine data first.

3. Describe one question that would be easier to answer with MongoDB than via a SQL query.

   If a question asking us to count or rank tweet replies after inserting more new user information. When database is set to grow big and data is location based, it's better to choose MongoDB.

4. Describe one question that would be easier to answer via a SQL query than using MongoDB.

   Just like question 3 in homework, if we want to find users with the most tweets, SQL is better than MongoDB. Because MongoDB need more time on aggregation and more complex process.

5. What fields in the original JSON structure would be difficult to convert to relational database schemas?

   The most difficult should be retweet_status, because it is recursive, which make the process more complex.

6. In terms of lines of code, when did various approaches shine? Think about the challenges of defining schemas, loading and storing the data, and running queries.

   When defining schemas, both SQL and protocol buffer need to re-encoding, however, MongoDB is schema-free, and spend 0 line of code.
   When loading and storing data, MongoDB could import json file very conveniently.
   When running queries, SQL is the most simple one to perform the analysis.
   Proto-buffers requires the most complexity on coding.

7. What other metrics (e.g., time to implement, code redundancy, etc.) can we use to compare these different approaches? Which system is better by those measures?

   Scalability-- if your database is dynamic, for example, data increases, MongoDB is better than SQL, for instance, if we want to add new user information frequently.

   Code and data readability—SQL is the best at readability of code and data structure. The table performance of SQL is really user-friendly.

   Time to implement—when we try to get the information we want, the time spend on coding is least for SQL, so SQL is the best at time to implement.

8. How long did this lab take you? We want to make sure to target future labs to not take too much of your time.

   It took 20 hours.