Danny Trinh
EC601
Professor Osama

Database Technologies

There are countless databases available for consumers and companies to use along side their products. But what is a database? A database is defined as a means for a user to store and access their data from their product. In addition to having the ability to store and access data, the user can perform administration tasks from anywhere they reside. This means that many database technologies are sometimes equipped with cloud capabilities. The following is a list of the most popular databases currently used by companies and consumers today: MongoDB, MySQL, Firebase, Redis, Cassandra. In this project, we will be comparing the various capabilities of database systems and demonstrating the use of MongoDB and MySQL.

The widely popular database MongoDB has been touted as the "NoSQL" database program. This statement has been used to the developers' advantage because SQL databases require users to have knowledge SQL syntax. However, security has always been a main concern for MongoDB. In the past, MongoDB databases have allowed all users access to information from their database. [1] As a result, tens of thousands of information have been held ransom. In version 3.6, any networked configured database must specifically be configured to allow users access bringing an end to this debacle. MongoDB has three main versions. The community version is free, the enterprise server is the commercial version and Atlas is the cloud capable version. Atlas can be run on AWS, Microsoft Azure, and Google Cloud. The selling points of MongoDB is that data is stored in JSON-like documents, ad hoc queries capable, and indexing and aggregation of data can be done in real time.

SQL by itself is a programming language and an acronym for Structured Query Language. It was originally developed by IBM in the early 1970's. The main syntax of SQL involves clauses, expressions, predicates, queries, and statements. The main SQL database used many individuals is MySQL. MySQL is owned by Oracle and their main users are Facebook, Twitter, and YouTube. [2] MySQL is globally renowned for being the most secure database management system which is why it is used by the largest of companies. It is also boasts round-the-uptime with many master/slave replication configurations, and specialized cluster server configurations. As a result, this means that it also scales well. Therefore, if a small company started out using MySQL, they can be assured that as they grow, their database will not become obsolete. Finally, since MySQL is owned by Oracle, the level of support available is another feature of MySQL.

Firebase is a mobile and web application platform that happens to also include a real-time database. Since Firebase is a mobile and web application platform, its database feature work to enhance a users' experience. From sharing photos in the cloud, to chatting with millions of users online, Firebase enables developers to quickly and easily implement these features into their applications. Firebase was recently acquired by Google, so the level of support is also available if needed. As one can see, the cloud capabilities are quite immense with Firebase and that is primarily the main feature of Firebase. Another advantage of using Firebase is their offline database. They allow developers to create apps with serverless databases optimized for offline use.

One of the most unique implementations of a database is by Redis Labs. Redis's database maps keys to various types of values. These can be list of strings, sets of strings, hash tables etc. Essentially

developers can map keys to any data structure. However, performance is not that great with Redis. It operates on a single process and is single threaded or double threaded. [3]

Setting up MongoDB to use was a breeze. The API allows the developer to create a server class and be up and running in minutes. The program that will be utilizing the MongoDB database will be the Flop or Not application from mini project 1. The Flop or Not application takes in a username, product and predicts whether or not a product is a good product or not. One thing to note is that the all collections are stored by default in /var/lib/mogodb. It was relatively simple to create a collection, add to the collection and query using JSON formats. The following is a screenshot of a Jupyter Notebook querying tweets and plotting a Gaussian distribution of the sentiment scores:

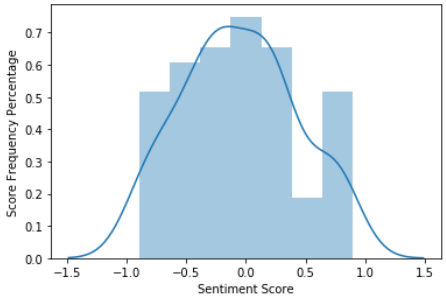
MongoDB Query and Usage

In [16]:
```
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mydatabase"]
mycol = mydb["tweets"]
mydb.mycol.remove({})
myquery = { "username": "Google" }

mydoc = mycol.find(myquery)

sent_score_list = []
for j in mydoc:
    sent_score_list.append(j['score'])
    #print(j['score'])
```

```
/usr/lib/python3/dist-packages/ipykernel_launcher.py:5: DeprecationWarning: remove is deprecated. Use delete_one or dele
te_many instead.
  """
```

In [22]:
```
x = np.array(sent_score_list)
g = sns.distplot(x);
g.set(xlabel='Sentiment Score', ylabel='Score Frequency Percentage')
```

Out[22]:
```
[Text(0, 0.5, 'Score Frequency Percentage'), Text(0.5, 0, 'Sentiment Score')]
```



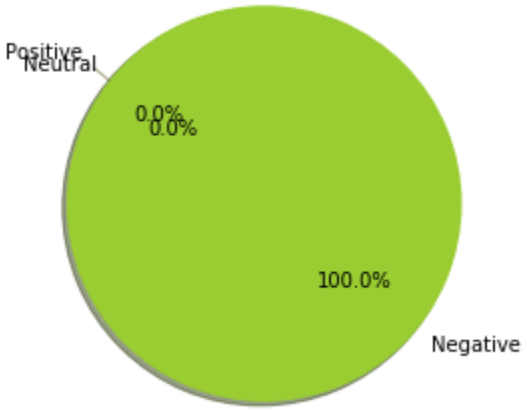
The following code finds the word Iphone in the storage of tweets using regular expressions:

Danny Trinh
EC601
Professor Osama

```
# query example for tweets containing word
import re

pat = re.compile(r'iphone', re.I)
mydoc = mycol.find({ "tweet": {'$regex': pat}})
pos_score = 0
neu_score = 0
neg_score = 0
for j in mydoc:
    score = float(j['score'])
    if score > 0:
        pos_score += 1
    elif score < 0:
        neg_score += 1
    else:
        neu_score += 1

    print(j)
```

```
{'_id': ObjectId('5dedc30259e9f90dd6df4df9'), 'username': 'Google', 'product': 'Pixel 4', 'score': -0.10000000149011612,
'tweet': 'That was shot on the iPhone 11 Pro in slowmo this is from the Pixel 4 from earlier and a different angle not i
n slowmo TeamPixel'}
{'_id': ObjectId('5dedc30659e9f90dd6df4e0c'), 'username': 'Google', 'product': 'Pixel 4', 'score': -0.8999999761581421,
'tweet': 'Well Google you finally did it I just embraced the other side By releasing a phone Pixel 4 XL with subpar spec
s mediocre battery I decided to go with the iPhone11ProMax google apple frompixel2iphone'}
{'_id': ObjectId('5dedc30659e9f90dd6df4e0d'), 'username': 'Google', 'product': 'Pixel 4', 'score': -0.699999988079071, '
tweet': 'Yap we are in the era of the ugly phone It started with the Google pixel 4 stove cameras then iPhone stove came
ras now Samsung bringing a bigger stove'}
```

We then plot scores with a pie chart to see how Iphone fairs in a Google Pixel 4 database.



It is all negative and reading over the tweets, we can conclude all the users switched to Iphones after hearing about the Google Pixel 4.

MySQL was a bit more involved to setup. It is very clear that security is the number one priority for MySQL. Users first have to setup a MySQL server on their machine. Afterwards, they have to start the server, setup the username and passwords that have access to the server then start the server. There are other Linux instructions in order to start up the server at boot. On the application side, the developer has to specify the user and password credentials within the program in order to access the server. After all that has been done, the learning curve is quite a bit steep because now developers must be familiar with the SQL programming language. Similarly to MongoDB, the database files are stored in /var/lib/mysql. The following is the equivalent MySQL call for querying a table.

Danny Trinh
EC601
Professor Osama

In [46]:
```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  passwd="",
  database="mydatabase"
)

mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM tweets")
myresult = mycursor.fetchall()
mydb.close()

sent_score_list = []
for j in myresult:
    sent_score_list.append(float(j[3]))
    print(j)
```

In conclusion, whether you are an app developer, or a multi-billion dollar company, or both, there is a database for you. From the creative use of storage to rock solid stability and security, databases are helpful for learning more about your users and to monitor the performance of your products. One of the biggest issues that stems from these databases growing number of server farms to store data. As we collect more and more data, storing all that information becomes an issue. From large power usage to memory shortages which drove up price in memory, the storage of data is another issue.

Danny Trinh
EC601
Professor Osama

References

[1] Krebs, Brian. "Extortionists Wipe Thousands of Databases, Victims Who Pay Up Get Stiffed". krebsonsecurity.com. Brian Krebs. Archived from the original on January 11, 2017. Retrieved January 11, 2017.

[2] Callaghan, Mark (13 April 2010). MySQL at Facebook. YouTube. Google. Retrieved 3 August 2010. x,000 servers, ... Master-slave replication, InnoDB

[3] "Redis on the Raspberry Pi: adventures in unaligned lands - <antirez>". antirez.com.