Music Databse

Sida Chen
001409406

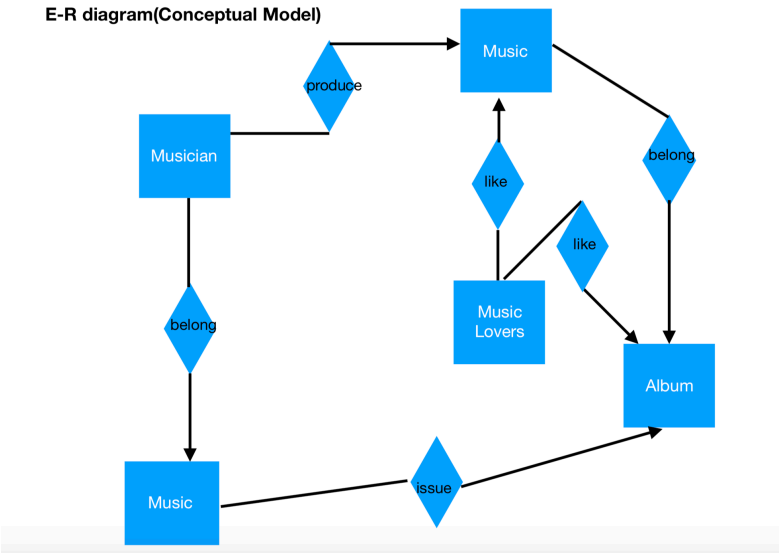
1.Introduction

In this final project, I aim to create a social data news site. As for the domain of my database, I plan to choose music. Therefore, in this domain, I decide to model some objects like music lovers, musicians, music company, album, music itself. Music lovers and musician correspond to people; music company corresponds to place, album and music correspond to things. All of words I mentioned above is my ideas about domain. These entities are basic structure of my database. In further work, these may be some tiny modifications for my database.

Next, I want to talk about attributes for each entity and model relationship between different entities. First of all, I want to identify initial attributes for each entity. For example, in music lovers entity, there must be some basic attributes like name, sex, birth, age, country, address, hobby and so on; in musicians entity, it should contain name, sex, birth, age, country, language and so on; in music company entity, it should contain name, address, rank and so on; in album entity, it should contain id, name, issued_date, sale, comment and so on; in music entity, it should contain name, issued_date, duration, language, style, lyrics, likes, comments and so on.

2.Model Database

At last, let's model the relationship between these entities, and the fact should be that musician produces music. Music belongs to album. Musician loves music. Musicians belongs to music company. Music lovers like music and album. Music company issues album. In the following, we are going to illustrate this conceptual model and relationship using E-R diagram.



2.1Gather Data

2.1.1 web script

In my project, I mainly using the 'BeautifulSoup' to do the web scripting.
Firstly, I set the url as the official website I need to script. And then I call the method. The next step is to try for each tag. I need to know what is the content of each tag. And I do some basic coding to find what is I really need.

Scraping musician lists of Wanerbros Records using BeautifulSoup

```
In [1]: from requests import get

In [2]: url = "http://www.warnerbrosrecords.com/artists"

In [3]: response = get(url)
        response

Out[3]: <Response [200]>

In [4]: from bs4 import BeautifulSoup as bs ## importing Beautiful Soup
        html_soup = bs(response.text,'html.parser')## python's in built library HTML parser
        type(html_soup)

Out[4]: bs4.BeautifulSoup

In [5]: singer_containers = html_soup.findAll('div',class_ ="view-content")

In [6]: print(len(singer_containers))
```

After I figure out what is the correct tag I need. I adopt use loop to display all of the information into the list. And save it into dataframe. Besides, I do some operation to re-set the index as "id".

```
In [20]: for i in range(len(b)):
             musician_id.append(i)
             music_company.append("Warnerbros Records")
             musican.append(b[i].li.div.text)

In [21]: musican

Out[21]: [' Andra Day ',
          ' Baka Not Nice ',
          ' Carlie Hanson ',
          ' Damon Albarn  ',
          ' Eric Clapton ',
          ' Feder ',
          ' Gallant ',
          ' Houndmouth ',
          ' Icarus ',
          ' Jason Derulo ',
```

```
In [28]: import pandas as pd
         music_company_df = pd.DataFrame({
                                 "musician_id":musician_id,
                                 "company_name":music_company,
                                 "musician":musican
                                })
         music_company_df
```

Out[28]:

	musician_id	company_name	musician
0	1	Warnerbros Records	Andra Day
1	2	Warnerbros Records	Baka Not Nice
2	3	Warnerbros Records	Carlie Hanson
3	4	Warnerbros Records	Damon Albarn

2.1.2 twitter API

The second method is to obtain data using Twitter API. At first, everyone need to create an account and request for the key to using twitter API.

The main function I used is 'twitter_api.search.tweets'. Set the name of musician as the keywords. And then we can get the related posts according to their name.

In [36]:
```
kuang = []
for q in musician:
    n=5
    search_results =  twitter_api.search.tweets(q=q,count=n)
    statuses = search_results['statuses']
    print(len(statuses))
    kuang.append(statuses)
```

This step is to save all the information in each tag to the list.

And the print the dataframe, save into .csv file.

In [42]:
```
import pandas as pd
for z in range(len(kuang)):
    for x in range(len(kuang[z])):
        music_lover_id.append(kuang[z][x]['user']['id'])
        music_lover_name.append(kuang[z][x]['user']['screen_name'])
        music_lover_followers_count.append(kuang[z][x]['user']['followers_count'])
        music_lover_friends_count.append(kuang[z][x]['user']['friends_count'])
        post_created_at.append(kuang[z][x]['created_at'])
        post_text.append(kuang[z][x]['text'])
        post_favorite_count.append(kuang[z][x]['favorite_count'])
        post_retweet_count.append(kuang[z][x]['retweet_count'])

        str1 = '#'
        for c in range(len(kuang[z][x]['entities']['hashtags'])):
            str1 += (kuang[z][x]['entities']['hashtags'][c]['text']) + '#'
        post_hashtags.append(str1)
        str2 = '@'
        for v in range(len(kuang[z][x]['entities']['user_mentions'])):
            str2 += (kuang[z][x]['entities']['user_mentions'][v]['screen_name']) + '@'
        post_mentions.append(str2)
```

In [43]:
```
music_lover_1_df = pd.DataFrame({
                        "music_lover_id":music_lover_id,
                        "music_lover_name":music_lover_name,
                        "music_lover_followers_count":music_lover_followers_count,
                        "music_lover_friends_count":music_lover_friends_count,
                        "post_created_at":post_created_at,
                        "post_text":post_text,
                        "post_hashtags": post_hashtags,
                        "post_mentions":post_mentions,
                        "post_favorite_count":post_favorite_count,
                        "post_retweet_count":post_retweet_count
                       })
music_lover_1_df
```

Out[43]:

	music_lover_id	music_lover_name	music_lover_followers_count	music_lover_friends_count	post_created_at	post_text	p
0	745341329136193539	kylie_colson	330	171	Sun Apr 21 03:36:47 +0000 2019	When Andra Day said "You're broken down and ti...	
1	1078050912944431104	kfaiplaylist	114	3	Sun Apr 21 02:39:04 +0000 2019	"Not Today" by Andra Day. Big Blue Train at 9:...	
2	533145152	Tobe_TheReason	707	538	Sun Apr 21 02:23:58 +0000 2019	Andra day- rise up 🙏🏾	
3	898078609243602944	villanelleve	202	121	Sun Apr 21 00:59:48 +0000 2019	larinha's top 5 artists this week: The xx (22)...	
4	69680902	Madabohki	1954	488	Sun Apr 21 00:20:33 +0000 2019	@OVOSound @Drake @Baka_Not_Nice https://t.co/i...	
					Sat Apr 20	BAKA NOT NICE on No	

In [44]:
```
indexed_music_lover_1_df = music_lover_1_df.set_index('music_lover_id')
indexed_music_lover_1_df
```

The bellowing screen-shots are how to get the "hashtags", "mentions".

Search for the entity['hashtags'] and get the text of the 'hashtags'.

Also, save the information into dataframe and csv.file.

In [50]:
```
musician_post_mentions=[]
musician_post_hashtags=[]
for tags in range(len(entities)):
    str3 = '#'
    str4 = '@'
    for num_tag in range(len(entities[tags]['hashtags'])):
        str3 += entities[tags]['hashtags'][num_tag]['text'] + '#'
    musician_post_hashtags.append(str3)

    for num_mention in range(len(entities[tags]['user_mentions'])):
        str4 += entities[tags]['user_mentions'][num_mention]['screen_name'] + '@'
    musician_post_mentions.append(str4)
```

In [51]:
```
musician_1_info_df = pd.DataFrame({
                        "musician_id":musician_id,
                        "musician_name":musician_name,
                        "musician_screen_name":musician_screen_name,
                        "musician_description":musician_description,
                        "musician_followers_count":musician_followers_count,
                        "musician_friends_count":musician_friends_count,
                       })
musician_1_info_df
```

Out[51]:

	musician_id	musician_name	musician_screen_name	musician_description	musician_followers_count	musician_friends_count
0	815537202	Andra Day	AndraDayMusic	Grammy nominated singer/songwriter. Signed to ...	370895	1660
1	513453504	Baka Not Nice	Baka_Not_Nice		8625	32
2	786784617596489728	Carlie Hanson	carliehanson	you are never alone 💙 BACK IN MY ARMS out 4/26...	6137	374
3	531750633	Damon Albarn	Damonalbarn	New album 'Merrie Land' from Damon Albarn and ...	274318	117

This part is to get the 'comments' information.

I call the 'api.search' method. And then I get the 'statues' tag. Here, I set the screen name I searched is the same as the name of the musician. So I can get the related information about comments.

In [68]:
```
import sys
replies=[]
mu_name = []
non_bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1), 0xfffd)
for mmmnnn in musician_screen_name:
    for full_tweets in tweepy.Cursor(api.user_timeline,screen_name=mmmnnn).items(10):
        for tweet in tweepy.Cursor(api.search,q='to:'+ mmmnnn,result_type='recent').items(10):
            if hasattr(tweet, 'in_reply_to_status_id_str'):
                if (tweet.in_reply_to_status_id_str==full_tweets.id_str):
                    replies.append(tweet.text)
                    mu_name.append(mmmnnn)
```

In [69]:
```
posts_reply_df = pd.DataFrame({
                        "musician_screen_name":mu_name,
                        "replies":replies
                       })
posts_reply_df
```

Out[69]:

	musician_screen_name	replies
0	carliehanson	@carliehanson @yungblud @Zhamakthecat cutiesss...
1	carliehanson	@carliehanson @yungblud i miss yall!!!
2	carliehanson	@carliehanson @yungblud @Zhamakthecat stop mak...
3	carliehanson	@carliehanson @yungblud @Zhamakthecat look lik...
4	carliehanson	@carliehanson @yungblud @Zhamakthecat my actua...
5	carliehanson	@carliehanson @yungblud @Zhamakthecat I miss b...

2.2 merge data

Merge different tables from 3 different tables into one table.

Because the original data are composed of three different companies. We need to merge three tables into one table. And this table represents the completed date in my database domain.

```
1 •  SELECT * FROM mydb.music;
2 •  INSERT INTO mydb.musician(musician_name,company_name)
3    SELECT musician,company_name FROM mydb.warnerbros_records_musicians;
4 •  INSERT INTO mydb.musician(musician_name,company_name)
5    SELECT musician,company_name FROM mydb.columbia_records_musicians;
6 •  INSERT INTO mydb.musician(musician_name,company_name)
7    SELECT musician,company_name FROM mydb.good_music_musicians;
```

2.3 clean data

Normalization
First normal form (1NF)
• Each table has a primary key: minimal set of attributes which can uniquely identify a record
• The values in each column of a table are atomic (No multi-value attributes allowed). all tables meet requirement
• There are no repeating groups: two columns do not store similar information in the same table.

Table 1: music_company

music_company_id	company_name
2	Columbia Records
3	Good Music
1	Warnerbros Records
NULL	NULL

Primary Key: music_company_id
This table meets all the requirements of 1NF.

Table 2: musician

musician_id	musician_name	company_name
1	After The Smoke	Warnerbros Records
2	Baka Not Nice	Warnerbros Records
3	Carlie Hanson	Warnerbros Records
4	Damon Albarn	Warnerbros Records
5	Emmylou Harris	Warnerbros Records
6	Faith Hill	Warnerbros Records
7	Gabrielle Aplin	Warnerbros Records
8	Houndmouth	Warnerbros Records
9	Icarus	Warnerbros Records
10	Jason Derulo	Warnerbros Records
11	K.D. Lang	Warnerbros Records
12	LCD Soundsyst…	Warnerbros Records
13	Mac Miller	Warnerbros Records

Primary Key: musician_id
This table meets all the requirements of 1NF.

Table 3: musician_tweet_info

musician_id	musician_name	musician_screen_name	musician_description	musician_followers_count	musician_friends_count
106854950	Celine Dion	celinedion	Posts signed / Publications signées TC = Team…	853160	95
1081032308	Emmylou Harris	EmmylouSongbird	Official news tweets for Emmylou Harris	54295	13
1152993973	Megan Bowen	MiGoOk_SaRaM	American🇺🇸in Seoul 🇰🇷 YouTuber/Vlogger 📹 F…	33053	125
116861193	Tony Williams	TWFTonyWilliams	Grammy-winning vocalist. New video #1 Fan is…	16550	957
1346180581	PARTYNEXTDOOR	partynextdoor	https://t.co/O58QdqcsOI	1464559	591
135019364	TRAVIS SCOTT	trvisXX	ASTROWORLD	5111912	727
1412296230	Icarus	icarus	Tom & Ian • Manager: ali@modestmanagement.…	4674	665
14791044	Faith Hill	FaithHill		1229631	4542
15553222	Consequence of Sound	consequence	Music, film, and pop culture publication.	197731	2297
156729608	Sabina Ddumba	SabinaDdumba		3058	416
1658490936	Ω	YusefMalik_	Quran [24:35]	197	239
16909755	LOVE	zenojones	@maniamerch ©1985-2018 Zeno Jones. ALL RI…	7749	1156
169686021	ye	kanyewest		29297295	269

Primary Key: musician_id

This table meets all the requirements of 1NF.

Table 4: musician_posts

post_id	musician_screen_name	p	post_text	post_hashtags	post_mentions	post_favorite_count	post_retweet_count
1007743020731547648	partynextdoor	F	@Ninja Great meeting…	#	@Ninja@	670	92
1010396186283393024	Neilyoung	S	Rolling to @arroyosec…	#potr#	@arroyosecowknd@	1444	252
1012603104791937024	partynextdoor	F	Scorpion out now @Dr…	#	@Drake@	29224	6562
1026787713893236736	Adele	T	https://t.co/PJVEFZm0…	#	@	16758	2662
1027215763734781952	SabinaDdumba	W	RT @Timrixlinden: @S…	#	@Timrixlinden@SabinaDdumba@RIXFM@	0	1
1028258409399943168	SabinaDdumba	S	KUNG FU KENNY 👑	#	@	5	0
1029049309487083521	Adele	M	https://t.co/1WDiyFHfN…	#	@	31602	4429
1031873144590675968	SabinaDdumba	T	RT @Dramaten: Sabin…	#appelgrenfriedner#	@Dramaten@SabinaDdumba@	0	1
1032249733027377152	SabinaDdumba	W	💛💖👧💖💛 https://…	#	@	37	5
1032271293226385408	SabinaDdumba	W	@p4tte TACK! 😇🙏…	#	@p4tte@	0	0
1032294401991688192	SabinaDdumba	W	@IngridSandraM Haha…	#	@IngridSandraM@	1	0
1036000485252063233	partynextdoor	S	RT @MannyWilkins5:…	#	@MannyWilkins5@	0	125
1036237308322824193	Damonalbarn	S	2nd September marks…	#	@	1987	284

Primary Key: post_id

This table meets all the requirements of 1NF.

Table 5: music

music_id	music _name	musician	album	company_id	duration_time
1	Rise Up	Andra Day	Cheers to the Fall	Warnerbros Records	04:13
2	Forever Mine	Andra Day	Cheers to the Fall	Warnerbros Records	03:19
3	Burn	Andra Day	The Hamilton Mixtape	Warnerbros Records	03:39
4	Only Love	Andra Day	Cheers to the Fall	Warnerbros Records	02:58
5	City Burns	Andra Day	Cheers to the Fall	Warnerbros Records	03:46
6	Baka Not Nice	Andra Day	Cheers to the Fall	Warnerbros Records	03:23
7	Live up to My Nme	Baka Not Nice	Baka Not Nice	Warnerbros Records	03:00
8	Dope Game	Baka Not Nice	4Milli	Warnerbros Records	03:39
9	My Town	Baka Not Nice	no long talk	Warnerbros Records	03:18
10	30	Baka Not Nice	no long talk	Warnerbros Records	03:12
11	Only One	Carlie Hanson	Only One	Warnerbros Records	03:10
12	Numb	Carlie Hanson	Numb	Warnerbros Records	02:13
13	Us	Carlie Hanson	Us	Warnerbros Records	03:19

Primary Key: music_id

This table meets all the requirements of 1NF.

Table 6: album

album_id	album_name	musician_name	released_year
1	4Milli	Baka Not Nice	2018
2	no long talk	Baka Not Nice	2019
3	Live Up to My Name	Baka Not Nice	2017
4	Cheers To The Fall	Andra Day	2015
5	Merry Christmas from Andra Day	Andra Day	2015
6	Only me	Carlie Hanson	2017
7	Numb	Carlie Hanson	2018
8	Why Did You Lie?	Carlie Hanson	2018
9	Mood	Carlie Hanson	2017
10	Everyday Robots	Damon Albarn	2014
11	Journey To The West	Damon Albarn	2008
12	Monkey Bee	Damon Albarn	2008
13	Profile	Emmylou Harris	1978

Primary Key: album_id
This table meets all the requirements of 1NF.

Table 7: music_lover_info

music_lover_id	music_lover_name	music_lover_followers_count	music_lover_friends_count
2800740257	MatixTulacz	177	687
560595571	deltatwelve	189	305
376818529	km_hard	278	223
43144848	D_In_Dopson	385	721
45515980	musicandblues	943	688
2773783732	solenecardon	304	172
1051327439836770306	19_21_25_Adele	207	25
3166608620	namturii	144	188
954611630	Nordetten	6	47
776463258110791680	stradiost11	2366	2533
2955431901	BaepsaeJo	814	662
776463258110791680	stradiost11	2366	2533
1069689098267283457	Wynndam1	90	1031

Primary Key: music_lover_id
This table meets all the requirements of 1NF.

Table 8: music_lover_tweet

music_lover_id	music_lover_name	post_created_at	post_text	post_hashtags	post_mentions	post_favorite_count	post_retweet_count
935170248419950592	seedlemosquito	Fri Apr 19 19:32:01 +0000 2019	RT @myday6official: i wan…	#	@myday6official@	0	19
111838120	LinosAvramides	Fri Apr 19 19:31:44 +0000 2019	AC/DC - Hells Bells (from…	#	@YouTube@	0	0
1682868955	KlncSlhn	Fri Apr 19 19:31:41 +0000 2019	Bu toplumun %60 DC ile A…	#	@	0	0
2373914336	ShibAvocate	Fri Apr 19 19:31:39 +0000 2019	@Adele_a_bout 😠	#	@Adele_a_bout@	0	0
179912821	mokonefinger	Fri Apr 19 19:31:35 +0000 2019	I think Daniel Caesar's Tin…	#	@	0	0
1650040938	wes_paiva	Fri Apr 19 19:30:53 +0000 2019	@PortalTracklist Taylor - A…	#	@PortalTracklist@	0	0
1123333886	adelumine	Fri Apr 19 19:30:53 +0000 2019	RT @lmaximus_e: No one…	#	@lmaximus_e@	0	1
1123333886	adelumine	Fri Apr 19 19:30:34 +0000 2019	RT @sugaryalien: no one:…	#	@sugaryalien@	0	2
1033011189196836864	BobOsFUN	Fri Apr 19 19:31:04 +0000 2019	It Pays to Play! One lucky…	#	@	0	0
172204354	thezoneplaylist	Fri Apr 19 19:21:02 +0000 2019	[April 19, 2019 at 12:19 p…	#	@	0	0
1026331723452231680	charmacrights	Fri Apr 19 19:13:18 +0000 2019	buzzcut season and the su…	#	@	3	0
2458374030	afinhy	Fri Apr 19 19:10:38 +0000 2019	Arcade Fire - Creature Co…	#	@	1	0
1083128334454845443	Cathimilena	Fri Apr 19 19:08:02 +0000 2019	@euWillianBonner wake u…	#	@euWillianBonne…	0	0

Primary Key: (music_lover_id, post_created_at)
This table meets all the requirements of 1NF.

Table 9: comments

musician_screen_name	comments
bmthofficial	@bmthofficial @allpointseastuk *Looking for rea…
bmthofficial	@bmthofficial @allpointseastuk @WarChildUK l…
bmthofficial	@bmthofficial i remember being 8 and scared s…
bmthofficial	@bmthofficial @scarlxrd Bro what is this shit wh…
celinedion	@celinedion @BSTHydePark @joshgroban @_…
ogchaseb	@ogchaseb https://t.co/D20p2yZIAu
ogchaseb	@ogchaseb All the Time!
chloexhalle	@chloexhalle @CBS went off!
chloexhalle	@chloexhalle @CBS Beautiful
chloexhalle	@chloexhalle @CBS yesss
chloexhalle	@chloexhalle @CBS So beautiful 😍
chloexhalle	@chloexhalle @CBS Can't wait loves!
chloexhalle	@chloexhalle @CBS So @RecordingAcad is d…

No Primary Key in comments table.
We need to add one primary key for this table. Here, we set the comments_id as its primary key and set it is auto increment.
The SQL statement is bellowing:

```
1      ALTER TABLE `mydb`.`comments`
2      ADD COLUMN `comment_id` INT NOT NULL AUTO_INCREMENT FIRST,
3      ADD PRIMARY KEY (`comment_id`);
4      ;
5
```

And the new table is like this:

comment_id	musician_screen_name	comments
1	bmthofficial	@bmthofficial @allpointseastuk *Looking for rea…
2	bmthofficial	@bmthofficial @allpointseastuk @WarChildUK l…
3	bmthofficial	@bmthofficial i remember being 8 and scared s…
4	bmthofficial	@bmthofficial @scarlxrd Bro what is this shit wh…
5	celinedion	@celinedion @BSTHydePark @joshgroban @_…
6	ogchaseb	@ogchaseb https://t.co/D20p2yZIAu
7	ogchaseb	@ogchaseb All the Time!
8	chloexhalle	@chloexhalle @CBS went off!
9	chloexhalle	@chloexhalle @CBS Beautiful
10	chloexhalle	@chloexhalle @CBS yesss
11	chloexhalle	@chloexhalle @CBS So beautiful 😍
12	chloexhalle	@chloexhalle @CBS Can't wait loves!
13	chloexhalle	@chloexhalle @CBS So @RecordingAcad is d…

Primary Key: comment_id
Right now, this table meets all the requirements of 1NF.

Second normal form (2NF)
- All requirements for 1st NF must be met.
- No partial dependencies.
- No calculated data

Table 1: music_company

Primary Key: company_id
This table meets all the requirements of 2NF.

Table 2: musician

Primary Key: musician_id
This table meets all the requirements of 2NF.

Table 3: musician_tweet_info

Primary Key: musician_id
This table meets all the requirements of 2NF.

Table 4: musician_posts

Primary Key: post_id
This table meets all the requirements of 2NF.

Table 5: music

music_id	music _name	musician	album	company_id	duration_time
1	Rise Up	Andra Day	Cheers to the Fall	Warnerbros Records	04:13
2	Forever Mine	Andra Day	Cheers to the Fall	Warnerbros Records	03:19
3	Burn	Andra Day	The Hamilton Mixtape	Warnerbros Records	03:39
4	Only Love	Andra Day	Cheers to the Fall	Warnerbros Records	02:58
5	City Burns	Andra Day	Cheers to the Fall	Warnerbros Records	03:46
6	Baka Not Nice	Andra Day	Cheers to the Fall	Warnerbros Records	03:23
7	Live up to My Nme	Baka Not Nice	Baka Not Nice	Warnerbros Records	03:00
8	Dope Game	Baka Not Nice	4Milli	Warnerbros Records	03:39
9	My Town	Baka Not Nice	no long talk	Warnerbros Records	03:18
10	30	Baka Not Nice	no long talk	Warnerbros Records	03:12
11	Only One	Carlie Hanson	Only One	Warnerbros Records	03:10
12	Numb	Carlie Hanson	Numb	Warnerbros Records	02:13
13	Us	Carlie Hanson	Us	Warnerbros Records	03:19

Primary Key: music_id
We notice that attribute company_id depends on musician.
Therefore, we need to drop this column.
If we want to know what company the musician belongs to, we can call table musician table and get right information we wanted.

SQL statement:

```
1  SELECT * FROM mydb.music;
2  ALTER TABLE music
3  DROP COLUMN company_id;
4  SELECT * FROM mydb.music;
```

100% 26:1

Result Grid Filter Rows: Search Edit:

music_id	music _name	musician	album	duration_time
1	Rise Up	Andra Day	Cheers to the Fall	04:13
2	Forever Mine	Andra Day	Cheers to the Fall	03:19
3	Burn	Andra Day	The Hamilton Mixtape	03:39
4	Only Love	Andra Day	Cheers to the Fall	02:58
5	City Burns	Andra Day	Cheers to the Fall	03:46
6	Baka Not Nice	Andra Day	Cheers to the Fall	03:23
7	Live up to My Nme	Baka Not Nice	Baka Not Nice	03:00
8	Dope Game	Baka Not Nice	4Milli	03:39
9	My Town	Baka Not Nice	no long talk	03:18
10	30	Baka Not Nice	no long talk	03:12

And then this table meets all the requirements of 2NF.

Table 6: album

Primary Key: album_id
This table meets all the requirements of 2NF.

Table 7: music_lover_info

Primary Key: music_lover_id
This table meets all the requirements of 2NF.

Table 8: music_lover_tweet

Primary Key: (music_lover_id, post_created_at)
This table meets all the requirements of 2NF.

Table 9: comments

Primary Key: comment_id
This table meets all the requirements of 2NF.

Third normal form (3NF)
- All requirements for 2nd NF must be met.
- Eliminate fields that do not directly depend on the primary key; that is no transitive

dependencies.

Table 1: music_company

Primary Key: company_id
This table meets all the requirements of 3NF.

Table 2: musician

Primary Key: musician_id
This table meets all the requirements of 3NF.

Table 3: musician_tweet_info

Primary Key: musician_id
This table meets all the requirements of 3NF.

Table 4: musician_posts

Primary Key: post_id
This table meets all the requirements of 3NF.

Table 5: music

Primary Key: music_id
This table meets all the requirements of 3NF.

Table 6: album

Primary Key: album_id
This table meets all the requirements of 3NF.

Table 7: music_lover_info

Primary Key: music_lover_id
This table meets all the requirements of 3NF.

Table 8: music_lover_tweet

Primary Key: (music_lover_id, post_created_at)
This table meets all the requirements of 3NF.

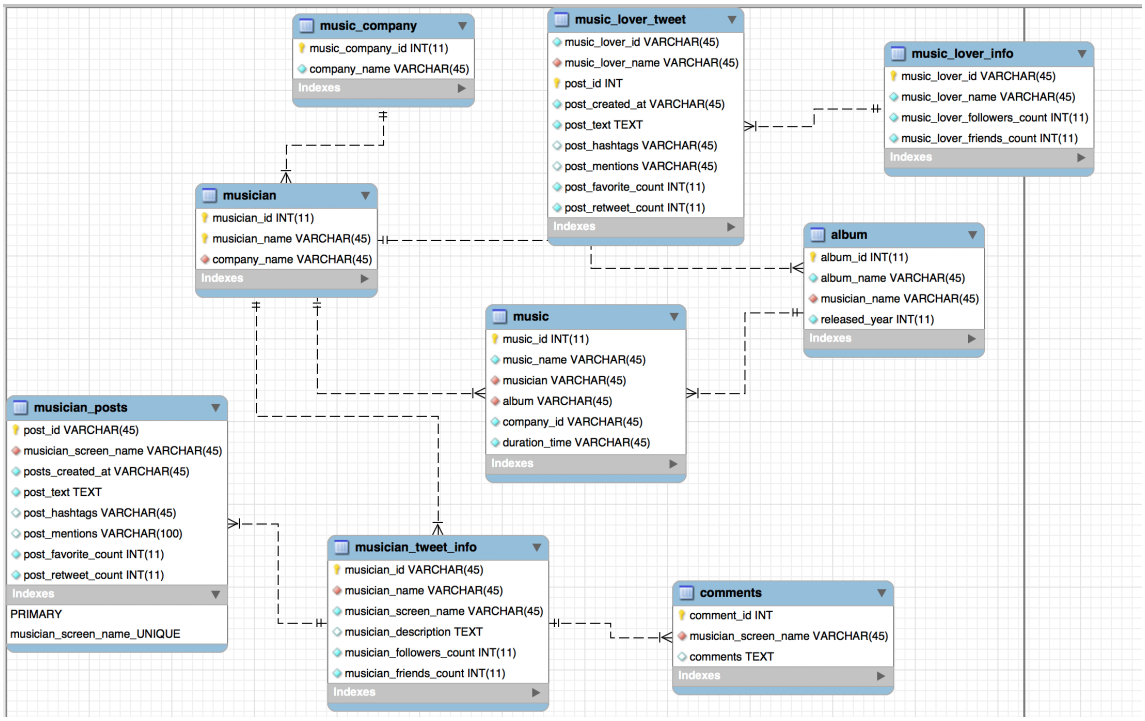
Table 9: comments

Primary Key: comment_id
This table meets all the requirements of 3NF.

2.4 Final Database

At last, my database is composed of 9 tables: music_company, musician, musician_tweet_info, musician_posts, music, album, music_lover_info, comments music_lover_posts.

Bellowing is my final completed EER (Enhanced Entity Relationships) Diagram:
Musician belongs to music_company. Each musician has it twitter account which contains the basic information of himself of herself. Musician_post responds to tweet_info.
Musician can compose a lot of music. Each music belongs to one album.
Music_lover can search something related musician or music. Music_lover_info contains the basic information of person. Each person can post some tweets.
At the same time, the comments table is related to musician_tweet_info. People can comments for the posts of musician. This is the whole relationship.



3.Analyze Questions

i. What are people saying about me (somebody)?

Imaging I am the 'jasonderulo'. If I want to know what people are saying to me. In the comment table, I need to get the comments. And the condition is that screen_name = 'my name'.

```
1 •  use mydb;
2
3 •  SELECT comments
4    FROM comments
5    WHERE musician_screen_name = 'jasonderulo';
6
```

100% 44:5

Result Grid | Filter Rows: Search | Export:

comments
@jasonderulo They really are not anymore, a S…
@jasonderulo @NICKIMINAJ @davidguetta 🔥…
@jasonderulo @NICKIMINAJ @davidguetta Du…
@jasonderulo @NICKIMINAJ @davidguetta 🔥…
@jasonderulo @Vante_BrookinsX @NICKIMIN…
@jasonderulo @NICKIMINAJ @davidguetta 🔥
@jasonderulo @NICKIMINAJ @davidguetta 🔥
@jasonderulo @KingRayMontanaa @NICKIMI…
@jasonderulo @NICKIMINAJ @davidguetta htt…
@jasonderulo @NICKIMINAJ @davidguetta 🔥…

ii. How viral are my posts?

As for how viral are my posts. It needs to count the total favorite counts and total retweet counts of one person.

```
7 •  SELECT musician_screen_name, SUM(post_favorite_count) AS total_favorite, SUM(post_retweet_count)AS total_retweet
8    FROM musician_posts
9    WHERE musician_screen_name = 'jasonderulo';
10
```

100% 6:5

Result Grid | Filter Rows: Search | Export:

musician_screen_name	total_favorite	total_retweet
jasonderulo	10234	1197

iii. What posts are likely to be interesting to me?

I choose the posts that contains my name. So the syntax is LIKE ('%my name%')

```
83 •  SELECT *
84    FROM music_lover_tweet
85    WHERE post_text LIKE ('%jason derulo%');
```

100% 1:78

Result Grid | Filter Rows: Search | Export:

music_lover_id	music_lover_name	post_created_at	post_text	post_hashtags	post_mentions	post_fav
442583944	RadioRideTheWav	Fri Apr 19 19:19:12 +0000 2019	(CLICK LINK TO LISTEN->) Now Playing: Jason Derulo - Talk Dirty…	#	@	0
1070158970780700672	Juan34637766	Fri Apr 19 19:15:01 +0000 2019	RT @pastadabs: A list of Celebrities/influencers that love blackpink: d…	#	@pastadabs@	0

iv. What posts are like mine?

For this question, firstly, I identify the object is posts. And then I need to consider the condition. Like mine means 'hashtags' is related to me. So I search for the hashtags

that are equals to my name.

```
18 •  SELECT post_text
19    FROM musician_posts
20    WHERE post_hashtags IN (
21    SELECT post_hashtags
22    FROM musician_posts
23    WHERE musician_screen_name = 'jasonderulo'
24    AND post_hashtags NOT IN ('#'))
25    AND musician_screen_name NOT IN ('jasonderulo');
```

100% 9:16

Result Grid | Filter Rows: Search | Export:

post_text
RT @PopCraveNet: #GameOfThrones will release their soundtrack on Friday, April 26th which features appearances from @TheWeeknd, @SZA, @Trvi...
RT @GameOfThrones: Tonight. #GameofThrones

v. What users post like me?

It is the same reason. Just the difference in the objects.

The object I need is the user. This attribute is also in the musician_posts table.

```
27 •  SELECT musician_screen_name
28    FROM musician_posts
29    WHERE post_hashtags IN (
30    SELECT post_hashtags
31    FROM musician_posts
32    WHERE musician_screen_name = 'jasonderulo'
33    AND post_hashtags NOT IN ('#'))
34    AND musician_screen_name NOT IN ('jasonderulo');
35
```

100% 49:34

Result Grid | Filter Rows: Search | Export:

musician_screen_name
chloexhalle
YusefMalik_

vi. Who should I be following?

I need to following people who has the same description with me. Because my domain is music, I need to follow people who also like music and who is musician. And then ‘group by’ followers_counts. I just want to follow people who has most numbers of followers.

```
36 •  SELECT musician_name, musician_followers_count
37    FROM musician_tweet_info
38    WHERE musician_description LIKE '%music%'
39    ORDER BY musician_followers_count DESC;
40
```

100% 1:35

Result Grid Filter Rows: Search Export:

musician_name	musician_followers_count
Calvin Harris	12789800
The Chosen One	1997083
Consequence of Sound	197731
CYHI THE PRYNCE	147675
Au/Ra	7626
ayokay	7219
KC DA BEATMONSTER	6784
Chris Blaze	392

vii. What topics are trending in my domain?

I just count how often each hashtags appears in my database.

And 'group by' the numbers. LIMIT 1 so that I can get the most popular trending.

```
42 •  SELECT post_hashtags,COUNT(*)
43    FROM musician_posts
44    WHERE post_hashtags NOT IN ('#')
45    GROUP BY post_hashtags
46    ORDER BY COUNT(*) DESC
47    LIMIT 1;
48
```

100% 1:40

Result Grid Filter Rows: Search

post_hashtags	COUNT(*)
#WorldsBest#	8

viii. What keywords/ hashtags should I add to my post?

Once my post has most numbers of 'favorite_counts', the hashtag is right thing I need to add.

```
50 •  SELECT post_hashtags
51    FROM musician_posts
52    WHERE post_favorite_count = (
53    SELECT MAX(post_favorite_count)
54    FROM musician_posts
55    WHERE post_hashtags NOT IN ('#'));
56
57
```

100% 1:48

Result Grid Filter Rows: Search Export:

post_hashtags
#NationalAdoptAShelterPetDay#

ix. Should I follow somebody back?

I want to judge whether the people have more 'followers_count' than me.
I just want to follow people who are more popular than me.

```
58 •  SELECT musician_name, musician_followers_count
59    FROM musician_tweet_info
60    WHERE musician_followers_count >=(
61    SELECT musician_followers_count
62    FROM musician_tweet_info
63    WHERE musician_screen_name = 'jasonderulo');
```

100% 12:55

Result Grid Filter Rows: Search Export:

musician_name	musician_followers_count
Adele	27804372
BEYONCÉ	14941669
Calvin Harris	12789800
Jason Derulo	3896739
ye	29297295
COMMON	5385573
John Legend	12623110
TRAVIS SCOTT	5111912
T-Raww	5375601

4.More Use Cases

1. Who should I be following?

```
36 •  SELECT musician_name, musician_followers_count
37    FROM musician_tweet_info
38    WHERE musician_description LIKE '%music%'
39    ORDER BY musician_followers_count DESC;
40
```

100% 1:35

Result Grid Filter Rows: Search Export:

musician_name	musician_followers_count
Calvin Harris	12789800
The Chosen One	1997083
Consequence of Sound	197731
CYHI THE PRYNCE	147675
Au/Ra	7626
ayokay	7219
KC DA BEATMONSTER	6784
Chris Blaze	392

2. What is the best time to post?

Find the post which has most favorite_count or retweet_count.
And get the 'created_at' attribute. The time is the best time to post.

```
66 •  SELECT posts_created_at, post_favorite_count, post_retweet_count
67    FROM musician_posts
68    ORDER BY post_favorite_count DESC, post_retweet_count DESC
69    LIMIT 1;
```

100% 1:65

Result Grid | Filter Rows: Search | Export: | Fetch rows:

posts_created_at	post_favorite_count	post_retweet_count
Tue Mar 05 19:28:02 +0000 2019	488054	60269

3. What's my reach?

Reach is composed of how many followers I have and how many friends I have.

```
71 •  SELECT musician_name, musician_followers_count, musician_friends_count, (musician_followers_count + musician_friends_count) AS reach
72    FROM musician_tweet_info
73    WHERE musician_screen_name = 'jasonderulo';
74
```

100% 9:69

Result Grid | Filter Rows: Search | Export: | Result Grid

musician_name	musician_followers_count	musician_friends_count	reach
Jason Derulo	3896739	13945	3910684

4. How many music have been issued for each music company?

Find numbers of music for each companies.

And use "UNION" to merge them into one table.

```
191 •  SELECT COUNT(musician_name),company_name
192    FROM music AS a JOIN musician AS b ON
193    a.musician = b.musician_name
194    WHERE b.company_name = 'Good Music'
195    UNION
196    SELECT COUNT(musician_name),company_name
197    FROM music AS a JOIN musician AS b ON
198    a.musician = b.musician_name
199    WHERE b.company_name = 'Columbia Records'
200    UNION
201    SELECT COUNT(musician_name),company_name
202    FROM music AS a JOIN musician AS b ON
203    a.musician = b.musician_name
204    WHERE b.company_name = 'Warnerbros Records';
205
```

35:204

Result Grid | Filter Rows: Search | Export:

COUNT(musician_name)	company_name
71	Good Music
57	Columbia Records
0	NULL

5. which company is most popular?

We need to consider the total followers of each company. Company has lots of musicians. The first thing is to count the total numbers of followers according to the classification of company. Here we create three views.

```
114 •   CREATE VIEW Columbia_Records_Fans AS
115     SELECT SUM(musician_followers_count) AS company_fans, company_name
116     FROM musician_tweet_info AS aa JOIN musician AS bb ON
117     aa.musician_name = bb.musician_name
118     AND bb.company_name = 'Columbia Records';
119
120 •   CREATE VIEW Warnerbros_Records_Fans AS
121     SELECT SUM(musician_followers_count) AS company_fans, company_name
122     FROM musician_tweet_info AS aa JOIN musician AS bb ON
123     aa.musician_name = bb.musician_name
124     AND bb.company_name = 'Warnerbros Records';
125
126 •   CREATE VIEW Good_Music_Fans AS
127     SELECT SUM(musician_followers_count) AS company_fans, company_name
128     FROM musician_tweet_info AS aa JOIN musician AS bb ON
129     aa.musician_name = bb.musician_name
130     AND bb.company_name = 'Good Music';
```

And then we UNION three tables because we can merge them together.
Use ‘order by’ and ‘LIMIT 1’ we can get the most popular one company.

```
145 •   CREATE VIEW joined_company_fans AS
146     SELECT *
147     FROM Good_Music_Fans
148     UNION
149     SELECT * FROM Columbia_Records_Fans
150     UNION
151     SELECT * FROM Warnerbros_Records_Fans;
152
153
154 •   SELECT *
155     FROM joined_company_fans
156     ORDER BY company_fans DESC
157     LIMIT 1;
158
159
160
```

32:106

Result Grid Filter Rows: Search Export: Fetch

company_fans	company_name
61936005	Columbia Records

5. VIEWS

1. Count the total fans(followers) of each company

```
114 •  CREATE VIEW Columbia_Records_Fans AS
115    SELECT SUM(musician_followers_count) AS company_fans, company_name
116    FROM musician_tweet_info AS aa JOIN musician AS bb ON
117    aa.musician_name = bb.musician_name
118    AND bb.company_name = 'Columbia Records';
119
120 •  CREATE VIEW Warnerbros_Records_Fans AS
121    SELECT SUM(musician_followers_count) AS company_fans, company_name
122    FROM musician_tweet_info AS aa JOIN musician AS bb ON
123    aa.musician_name = bb.musician_name
124    AND bb.company_name = 'Warnerbros Records';
125
126 •  CREATE VIEW Good_Music_Fans AS
127    SELECT SUM(musician_followers_count) AS company_fans, company_name
128    FROM musician_tweet_info AS aa JOIN musician AS bb ON
129    aa.musician_name = bb.musician_name
130    AND bb.company_name = 'Good Music';
```

```
133 • SELECT *
134   FROM Columbia_Records_Fans,Warnerbros_Records_Fans,Good_Music_Fans;
135
```

68:134

Result Grid | Filter Rows: Search | Export:

company_fans	company_name	company_fans	company_name	company_fans	company_name
61936005	Columbia Records	NULL	NULL	23602292	Good Music

2. Count all music issued by each company

```
206 •  CREATE VIEW Columbia_Records_Music AS
207    SELECT music_name, musician
208    FROM music AS a JOIN musician AS b ON
209    a.musician = b.musician_name
210    WHERE b.company_name = 'Columbia Records';
211
212 •  CREATE VIEW Warnerbros_Records_Music AS
213    SELECT music_name, musician
214    FROM music AS a JOIN musician AS b ON
215    a.musician = b.musician_name
216    WHERE b.company_name = 'Warnerbros Records';
217
218 •  CREATE VIEW Good_Music_Music AS
219    SELECT music_name, musician
220    FROM music AS a JOIN musician AS b ON
221    a.musician = b.musician_name
222    WHERE b.company_name = 'Good Music';
```

```
224 •   SELECT *
225     FROM Columbia_Records_Music;
226
227 •   SELECT *
228     FROM Warnerbros_Records_Music;
229
230 •   SELECT *
231     FROM Good_Music_Music;
232
```

29:221

Result Grid | Filter Rows: Search | Export

music_name	musician
Miexed Personalities	Kanye West
Heartless	Kanye West
All Mine	Kanye West
Black Skinhead	Kanye West
Yikes	Kanye West
POWER	Kanye West

6. Functions

Function1:

Input is the name of musician. This functions returns the all of the music composed by this musician.

```
1 •  CREATE DEFINER=`root`@`localhost` FUNCTION `GetMusic`(N VARCHAR(45)) RETURNS varchar(255) CHARSET utf8mb4
2        READS SQL DATA
3        DETERMINISTIC
4    BEGIN
5    DECLARE qqq VARCHAR(255);
6     SELECT music_name INTO qqq FROM music WHERE musician = N;
7     RETURN qqq;
8
9    END
```

Function2:

Input is the id of musician. This functions returns the total numbers of the music composed by this musician.

```
1 •  CREATE DEFINER=`root`@`localhost` FUNCTION `get_music_num`(N VARCHAR(45)) RETURNS int(11)
2        READS SQL DATA
3        DETERMINISTIC
4    BEGIN
5    DECLARE num VARCHAR(255);
6     SELECT COUNT(music_name) INTO num FROM music WHERE musician = N;
7     RETURN num;
8
9    END
```

7. Stored Procedures

Procedure1:
Input is the id of musician. This functions returns the basic music information of this specific musician.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_musician_company`(IN FIRST INT)
BEGIN
 DECLARE musicianId INT;

    SET musicianId = FIRST;

 select c.name,m.musicianId, m.musician_name
    from `musician` as m
    join `music` as c on c.musician = m.musician_name
    where m.musician_id = musicianId;

 END
```

Procedure2:
Input is the id of musician. This functions returns the basic twitter information of this specific musician.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_musician_twitter_count`(IN FIRST INT)
BEGIN
 DECLARE postID INT;

    SET postID = FIRST;

 select m.musician_id,m.musician_name,m.musician_screen_name, m.musician_description, m.musician_followers_count, m.musician_friends_count
    from `musician_tweet_info` as m
    join `musician_posts` as t on m.musician_screen_name = t.musician_screen_name
    where t.post_id = postID;

END
```

8. Hashtags

8.1 Domain tags

```
In [82]: total_hashtags
Out[82]: ['SITWFest',
          'LeBatardAF',
          'AvengersEndgame',
          'listing',
          'Rutherford',
          'NJ',
          'realestate',
          'Alamzaibpashtunhero',
          'Alamzaibpashtunhero',
          'HipHopDX',
          'LOOSE',
          'dremstuff',
          'Gotham',
          'NP',
          'GGRL',
          'OURMUSIC',
          'OURCULTURE',
          'Mercedes',
          'AMG',

In [83]: len(total_hashtags)
Out[83]: 160
```

```
1 • SELECT * FROM mydb.hashtags;
```

100% 1:1

Result Grid Filter Rows: Search Ec

id	hashtags
1	SITWFest
2	LeBatardAF
3	AvengersEndgame
4	listing
5	Rutherford
6	NJ
7	realestate

8.2 Synonyms

```
In [87]: hashtags_synsets = []
         wn.synsets('WorldWaterDay')
         wn.synset('water.n.01').lemma_names
         for words in (wn.synset('water.n.01').lemma_names()):
             hashtags_synsets.append(words)
         hashtags_synsets

Out[87]: ['water', 'H2O']

In [88]: wn.synsets('CleanWaterHere2019')
         wn.synset('clean.n.01').lemma_names
         for words in (wn.synset('clean.n.01').lemma_names()):
             hashtags_synsets.append(words)
         hashtags_synsets

Out[88]: ['water', 'H2O', 'clean_and_jerk', 'clean']
```

I call the ‘wordnet’ package and do some coding operations.

```
1 • SELECT * FROM mydb.synonyms_hashtags;
```

100% 1:1

Result Grid Filter Rows: Search Edit:

id	hashtags
1	water
2	H2O
3	clean_and_jerk
4	clean
5	sale
6	Friday
7	Fri
8	rock
9	stone
10	trip
11	school

8.3 Mis-spelling

```
1 • SELECT * FROM mydb.mis_spelling;
```

100% 1:1

Result Grid Filter Rows: Search Edit:

id	hashtags
1	DC
2	accountability
3	humanrights

274	goddamn
275	DX
276	DZ
277	DV
278	DB
NULL	NULL

mis_spelling 1

9. Database Display

Step 1: Connect to my database

```
In [1]: import pymysql

In [2]: conn=pymysql.connect(host='localhost',user='root',passwd='11111111',db='mydb',port=3306)
        cur=conn.cursor()
```

Step 2: Execute SQL Statements

```
In [3]: cur.execute('select * from music_company')
Out[3]: 3

In [4]: D = cur.fetchall()
```

Step 3: if-else condition to display all the information in my database

```
In [*]: print('####Please input the table name that you want to search in databse:  ####')
name = input()
name = str(name)
if(name == 'music'):
    cur.execute('select * from music')
    D = cur.fetchall()
    print('id         ','music_name         ','musician          ','album          ','company         ',' time')
    for d in D:
        print(d)

elif(name == 'musician'):
    cur.execute('select * from musician')
    D = cur.fetchall()
    for d in D:
        print(d)

elif(name == 'music_company'):
    cur.execute('select * from music_company')
    D = cur.fetchall()
    for d in D:
        print(d)

elif(name == 'musician_tweet_info'):
    cur.execute('select * from musician_tweet_info')
    D = cur.fetchall()
    for d in D:
        print(d)

elif(name == 'musician_posts'):
    cur.execute('select * from musician_posts')
    D = cur.fetchall()
    for d in D:
        print(d)

elif(name == 'album'):
    cur.execute('select * from album')
```

10. Results

I have designed a completed database about music. And I meet all of the requirements in the project. In this process, I learned a lot such as the basic operation about python, basic knowledge of the NLP. Besides, I am going to have a deeper understanding of database. This final project is an exercise for me to challenge myself. I met a lot of challenges and with the help of many people I did it. This course is very important for me to learn more about data science.

11. Reference

https://twitter.com/AppleMusic

https://www.grammy.com https://www.youtube.com/feed/trending?bp=4gIuCggvbS8wNHJsZhIiUExGZ3F1TG5MNTlhbVBud2pLbmNhZUp3MDYzZlU1M3Q0 cA%3D%3D

https://www.guru99.com/functions.html

https://developer.twitter.com/en/docs.html

https://www.geeksforgeeks.org/nlp-wordnet-for-tagging/

LICENSE