Assignment 3

Yijun Zhou #52533998

In [1]:
```
import pandas as pd
import numpy as np
from sqlalchemy import create_engine
import sqlalchemy
```

Collect Data from Twitter API

In [2]:
```
#Keys for Twitter API
CONSUMER_KEY = "ZNLSW0nyWYv5fgEQ3eawyBj8z"
CONSUMER_SECRET = "1KlXyWjeCMqSbGNuvQeYcFj2pYawN6GQNZNfhXP3hGZjvyPzla"
ACCESS_KEY = "3309288092-yxGwoIzFDBRhljkEgpCgLfwF0MAwLyhqkXGhQgm"
ACCESS_SECRET = "1sioTIWicgaTacW3T6zXpxAd25D41LzdP2tXTMr6EYOCZ"
```

In [3]:
```
import tweepy

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)

api = tweepy.API(auth)
```

In [50]:
```
def collectData(query,item_num):
    data_results = []
    for tweet in tweepy.Cursor(api.search_tweets,
                               q=query,
                               result_type="recent",
                               include_entities=True,
                               lang="en").items(item_num):
        data_results.append(tweet)
    print("List lenghth:" + str(len(data_results)))
    return data_results
```

In [52]:
```
data_B1 = collectData("Bridgerton",2500)
```

```
List lenghth:2500
```

In [54]:
```
data_B2 = collectData("bridgerton netflix",2500) #9:31
```

```
List lenghth:2500
```

In [65]:
```
data_B3 = []
for tweet in tweepy.Cursor(api.search_tweets,
                           q='bridgerton',
                           until="2022-01-30",
                           include_entities=True,
                           lang="en").items(2500):
    data_B3.append(tweet)
print("List lenghth:" + str(len(data_B3)))
```

```
List lenghth:2500
```

In [56]:
```
data_Q1 = collectData("The Queen's Gambit",2500)
```

```
List lenghth:2441
```

In [58]:
```
data_Q2 = collectData("Beth Harmon",2500)
```

```
List lenghth:426
```

In [80]:
```
data_Q3 = collectData("The Queen's Gambit",2500)
```

```
List lenghth:2459
```

In [88]:
```
data_Q4 = collectData("The Queen's Gambit",2500)
```

```
List lenghth:2456
```

In [66]:
```
data_Bridgerton = set(data_B1 + data_B2 + data_B3)
print(len(data_Bridgerton))
```

```
5009
```

In [90]:
```
data_Queen = set(data_Q1 + data_Q2 + data_Q3 + data_Q4)
print(len(data_Queen))
```

```
2866
```

In [91]:
```
data_all = data_Bridgerton.union(data_Queen)
print(len(data_all))
```

```
7867
```

Transform Data into Dataframe

In [97]:
```
##transform functions:

def getTweets(tweet):
    trec = {"created_at": tweet.created_at,#time.mktime(tweet.created_at.timetuple()),
            "tweetid":tweet.id,
            "userid":tweet.user.id,
            "text":tweet.text,
            "in_reply_to_status_id":tweet.in_reply_to_user_id,
            "raw_tweet":tweet._json}
    return trec

def getStaticUserInfo(user):
    trec = {"userid":user.id,
            "username":user.name,
            "screen_name":user.screen_name,
            "userdescription":user.description}
    return trec

def getDynamicUserInfo(user):
    trec = {"created_at":user.created_at,
            "userid":user.id,
            "statuses_count":user.statuses_count,
            "followers_count":user.followers_count}
    return trec

def getEntityInfo(tweet):
    trec = {"tweetid":tweet.id,
            "hashtags":tweet.entities['hashtags']}
    return trec
```

In [76]:
```
##Postgres connnection info
url = 'postgresql+psycopg2://{}:{}@{}:{}/{}'.format('postgres','2245','localhost','5432','stats170a
engine = create_engine(url)
```

BridgertonTweetInfo

In [75]:
```
Bridgerton = [getTweets(pt) for pt in data_Bridgerton]

df_Bridgerton = pd.DataFrame(Bridgerton).drop_duplicates(['tweetid'])

print(len(df_Bridgerton))
df_Bridgerton.head()
```

```
5009
```

Out[75]:

	created_at	tweetid	userid	text	in_reply_to_status_id	raw_tweet
0	2022-02-02 03:52:00+00:00	1488721839887384576	1604249562	RT @Anladybee: Kate Sharma in Bridgerton Seaso...	NaN	{'created_at': 'Wed Feb 02 03:52:00 +0000 2022...
1	2022-01-28 23:08:15+00:00	1487200881460215808	3316300908	@Christopher_NLC I'm currently reading An Offe...	1.821024e+08	{'created_at': 'Fri Jan 28 23:08:15 +0000 2022...
2	2022-02-01 02:23:34+00:00	1488337199313551362	1344867125790298112	RT @chararirii: The Bridgerton. https://t.co/m...	NaN	{'created_at': 'Tue Feb 01 02:23:34 +0000 2022...
3	2022-01-28 20:19:53+00:00	1487158508017516544	1305937228783181824	How rich is the Bridgerton family? *runs to Go...	NaN	{'created_at': 'Fri Jan 28 20:19:53 +0000 2022...
4	2022-01-28 17:18:36+00:00	1487112889550864384	742703809235804162	everyone involved in bridgerton finds my suffe...	7.427038e+17	{'created_at': 'Fri Jan 28 17:18:36 +0000 2022...

In [79]:
```
##insert table
df_Bridgerton.to_sql("bridgertontweetinfo", con = engine, if_exists = 'replace', index = False,dtype
                {'raw_tweet':sqlalchemy.types.JSON}, schema = 'twitter')
```

Out[79]:
```
9
```

QueenTweetInfo

In [84]:
```
Queen = [getTweets(pt) for pt in data_Queen]

df_Queen = pd.DataFrame(Queen).drop_duplicates(['tweetid'])

print(len(df_Queen))
df_Queen.head()
```

```
2866
```

Out[84]:

	created_at	tweetid	userid	text	in_reply_to_status_id	raw_tweet
0	2022-02-02 04:21:03+00:00	1488729150068244480	993748794880671745	@meleesadposts Is that the guy from the Queen'...	3.404476e+09	{'created_at': 'Wed Feb 02 04:21:03 +0000 2022...
1	2022-01-31 05:16:14+00:00	1488018261627502593	812424309285105664	RT @AcademiaDreams: The Queen's Gambit (2020) ...	NaN	{'created_at': 'Mon Jan 31 05:16:14 +0000 2022...
2	2022-02-02 04:44:20+00:00	1488735010827870210	1306319934528475136	they should have ended the queen's gambit with...	NaN	{'created_at': 'Wed Feb 02 04:44:20 +0000 2022...
3	2022-02-02 04:30:45+00:00	1488731593217015810	751022622373191684	RT @tasty: Sit down and watch The Queen's Gamb...	NaN	{'created_at': 'Wed Feb 02 04:30:45 +0000 2022...
4	2022-02-02 03:27:56+00:00	1488715784578867204	1079563520478453760	@laisassauro the queen's gambit	1.473370e+18	{'created_at': 'Wed Feb 02 03:27:56 +0000 2022...

In [86]:
```
##insert table
df_Queen.to_sql("queentweetinfo", con=engine, if_exists='replace', index=False,dtype =
                {'raw_tweet':sqlalchemy.types.JSON}, schema = 'twitter')
```

Out[86]:
```
866
```

StaticUserInfo

In [98]:
```
Static = [getStaticUserInfo(pt.user) for pt in data_all]

df_Static = pd.DataFrame(Static).drop_duplicates(['userid'])

print(len(df_Static))
df_Static.head()
```

```
6128
```

Out[98]:

	userid	username	screen_name	userdescription
0	1604249562	-,`amiˀ,´-	amikosmos	InternationalPopKSensationSunshineRainbowTradi...
1	3316300908	dark_princess000	darkprincess_00	
2	1344867125790298112	delilah	dgeyoydelilah	
3	1305937228783181824	A Fezco Apologist	sameolegal	🕯️manifesting Carter x Taylor🕯️
4	742703809235804162	so you find my smile pleasing	connellschains	i will always seek to make it summer for you

In [99]:
```
##insert table
df_Static.to_sql("staticuserinfo", con=engine, if_exists='append', index=False, schema = 'twitter')
```

Out[99]:
```
128
```

DynamicUserInfo

In [101…
```
Dynamic = [getDynamicUserInfo(pt.user) for pt in data_all]

#get the lastest dynamicinfo in collected dataset
df_Dynamic = pd.DataFrame(Dynamic).sort_values(by='created_at', ascending=False).drop_duplicates(['u
```

```
df_Dynamic = df_Dynamic.iloc[:,1:4]
print(len(df_Dynamic))
df_Dynamic.head()
```

```
6128
```

Out[101]:

	userid	statuses_count	followers_count
4820	1488584836084342786	46	0
6016	1488577819458342913	7	3
7826	1488575055290380331	8	0
6431	1488571885122056194	7	4
7709	1488476516090626050	1	3

In [103…

```
##insert table
df_Dynamic.to_sql("dynamicuserinfo", con=engine, if_exists='append', index=False, schema = 'twitter
```

Out[103]: 128

TweetEntity

In [106…

```
Entity = [getEntityInfo(pt) for pt in data_all]

df_Entity = pd.DataFrame(Entity)

##make the table satisfy 1NF
arr_hashtags = []
for i in df_Entity.hashtags:
    if len(i) == 0:
        arr_hashtags.append(i)
    else:
        temp=[]
        for j in range(len(i)):
            temp.append(i[j]['text'])
        arr_hashtags.append(temp)
arr_hashtags

df_Entity_1NF = df_Entity.loc[df_Entity.index.repeat(df_Entity.hashtags.str.len())].assign(hashtags=
df_Entity_1NF.rename(columns={'hashtags':'hashtag'}, inplace = True)

print(len(df_Entity_1NF))
df_Entity_1NF.head()
```

```
2287
```

Out[106]:

	tweetid	hashtag
8	1488249503865995265	booktwt
8	1488249503865995265	booktwitter
29	1487122067543625731	bridgerton
29	1487122067543625731	ladywhistledown
35	1487512318917361670	bridgerton

In [107…

```
##insert table
df_Entity_1NF.to_sql("tweetentity", con=engine, if_exists='append', index=False, schema = 'twitter'
```

Out[107]: 287

In [ ]:

Part 2

```sql
SET search_path TO twitter;

CREATE TABLE IF NOT EXISTS BridgertonTweetInfo
(
    created_at              timestamp NOT NULL,
    tweetid                 bigint    NOT NULL,
    userid                  bigint    NOT NULL,
    text                    varchar,
    in_reply_to_status_id   bigint,
    raw_tweet               jsonb,
    PRIMARY KEY (tweetid),
    FOREIGN KEY (userid) references StaticUserInfo
);

CREATE TABLE IF NOT EXISTS QueenTweetInfo
(
    created_at              timestamp NOT NULL,
    tweetid                 bigint    NOT NULL,
    userid                  bigint    NOT NULL,
    text                    varchar,
    in_reply_to_status_id   bigint,
    raw_tweet               jsonb,
    PRIMARY KEY (tweetid),
    FOREIGN KEY (userid) references StaticUserInfo
);
```

```sql
CREATE TABLE IF NOT EXISTS StaticUserInfo
(
    userid                  bigint NOT NULL,
    username                varchar,
    screen_name             varchar,
    userdescription         varchar,
    PRIMARY KEY (userid)
);

CREATE TABLE IF NOT EXISTS DynamicUserInfo
(
    userid                  bigint NOT NULL,
    statuses_count          bigint,
    followers_count         bigint,
    PRIMARY KEY (userid),
    FOREIGN KEY (userid) references StaticUserInfo
);

CREATE TABLE IF NOT EXISTS TweetEntity
(
    tweetid             bigint   NOT NULL,
    hashtag             varchar
);
```

Part 3

a)

Observation:

I think most of the top words are neutral for both shows. There are many words are common stop words in English and neutral nouns . Other than that, for Bridgerton, there are three positive words detected: “like”, “love” and “kindly”. For the Queen's Gambit, negtive and postive word are both detected, such as “deny” and “like”.

Code:

```sql
--For Bridgerton:
SELECT word,
    count(1)::float/(select count(1) from BridgertonTweetInfo)::float as occurrence_rate
FROM BridgertonTweetInfo
    ,UNNEST(regexp_split_to_array(
                left(text,position('https://'in text)-1) --ignore the urls in tweet
                ,'[^A-Za-z0-9@#]') -- split by special characters
            ) AS word
where word not in('','RT')
    AND length(word)>=3 --only keep sequnece of 3 or more alphanumeric characters
GROUP BY word
ORDER BY occurrence_rate DESC
LIMIT 100;

--For the Queen's Gambit
SELECT word,
    count(1)::float/(select count(1) from QueenTweetInfo)::float as occurrence_rate
FROM QueenTweetInfo
    ,UNNEST(regexp_split_to_array(
                left(text,position('https://'in text)-1) --ignore the urls in tweet
                ,'[^A-Za-z0-9@#]') -- split by special characters
            ) AS word
where word not in('','RT')
    AND length(word)>=3 --only keep sequnece of 3 or more alphanumeric characters
GROUP BY word
ORDER BY occurrence_rate DESC
LIMIT 100;
```

Output for Bridgerton：

	word text	occurrence_rate double precision
1	Bridgerton	0.43960870433220206
2	the	0.38153127610301455
3	for	0.18925933320023958
4	and	0.1780794569774406
5	#Bridgerton	0.12377720103813136
6	bridgerton	0.12317827909762427
7	Nicola	0.11399480934318228
8	her	0.11339588740267519
9	Coughlan	0.11060091834697544
10	you	0.09502894789379118
11	star	0.09023757236973448
12	The	0.08704332202036334
13	season	0.08464763425833499
14	body	0.07306847674186465
15	this	0.06787781992413655
16	@chaoticguitar	0.06687961668995808
17	about	0.06508285086843682
18	@NoContxtBR	0.06288680375324417
19	series	0.05709722499500899
20	with	0.05350369335196646
21	like	0.05210620882411659
22	Kate	0.04791375524056698
23	was	0.0467159113595528
24	coming	0.046316630065881416
25	#bridgerton	0.045717708125374325
26	that	0.04451986424436015
27	can	0.043322020363345974
28	looking	0.042124176482331806
29	Season	0.04192453583549611
30	are	0.04132561389498902
31	@imnicjuarez	0.04032741066081054
32	following	0.04032741066081054
33	writers	0.04012777001397484
34	Snowfall	0.04012777001397484
35	@vulture	0.04012777001397484
36	Bridge	0.03992812936713915
37	culture	0.039728488720303455
38	assign	0.03952884807346776
39	Ackman	0.03952884807346776
40	Bill	0.03952884807346776
41	Atlanta	0.03952884807346776
42	recaps	0.03952884807346776
43	Anthony	0.038929926132960674
44	stop	0.03793172289878219
45	Netflix	0.03693351966460371
46	just	0.03673387901776802
47	but	0.03673387901776802
48	your	0.03573567578358954
49	BRIDGERTON	0.03533639448991815
50	people	0.03473747254941106
51	see	0.033140347374725494
52	Sharma	0.0329407067278898
53	opinions	0.03194250349371132
54	commenting	0.030744659612697144
55	have	0.030744659612697144
56	has	0.03054501896586145
57	not	0.03014573767219006
58	would	0.029746456378518665
59	@charariri	0.02954681573168297
60	from	0.029147534438011578
61	2022	0.028548612497504493
62	all	0.02715112796965462
63	one	0.026552206029147533
64	love	0.02635256538231184
65	had	0.02635256538231184
66	out	0.026152924735476143
67	THE	0.025753643441804752
68	fans	0.025554002794969055
69	being	0.02535436214813336
70	@Anladybee	0.024555799560790577
71	BALL	0.024156518267119186
72	days	0.024156518267119186
73	she	0.02395687762028349
74	get	0.023158315032940708
75	until	0.02295867438610501
76	wait	0.022759033739269314
77	Julia	0.022759033739269314
78	real	0.021760830505090836
79	two	0.021760830505090836
80	time	0.021760830505090836
81	everyone	0.021561189858255142
82	characters	0.021361549211419445
83	kept	0.020962267917748054
84	will	0.020962267917748054
85	2020	0.020962267917748054
86	yourself	0.020762627270912357
87	Her	0.020562986624076664
88	@BuzzFeed	0.020562986624076664
89	month	0.020562986624076664
90	Daphne	0.020363345977240967
91	kindly	0.02016370533040527
92	when	0.019964064683569576
93	@KennethDredd	0.01976442403673388
94	SHUT	0.01976442403673388
95	appreciate	0.01976442403673388
96	Hell	0.01976442403673388
97	his	0.019564783389898182
98	Body	0.019564783389898182
99	Quinn	0.019564783389898182
100	finally	0.018965861449391094

Output for the Queen's Gambit：

	word text	occurrence_rate double precision
1	The	0.7107466852756456
2	Queen	0.6577110956036287
3	Gambit	0.6273551988834613
4	the	0.4124214933705513
5	Netflix	0.27983251919050944
6	for	0.2198185624563852
7	defamation	0.20341939986043267
8	chess	0.16922540125610608
9	and	0.15561758548499652
10	after	0.15073272854152128
11	2020	0.12630844382414516
12	@AcademiaDreams	0.12037683182135381
13	suing	0.10711793440334962
14	she	0.10432658757850663
15	queen	0.10432658757850663
16	that	0.10293091416608513
17	grandmaster	0.10118632240055828
18	this	0.10013956734124214
19	show	0.09979064898813678
20	beth	0.09909281228192603
21	harmon	0.09909281228192603
22	first	0.09769713886950454
23	never	0.09665038381018842
24	woman	0.0942079553384508
25	gambit	0.0928122819260293
26	claimed	0.09176552686671319
27	named	0.09106769016050244
28	@ajplus	0.09071877180739707
29	fac	0.0903698534542917
30	lawsuit	0.08653175157013258
31	judge	0.07920446615491974
32	dismiss	0.07641311933007676
33	motion	0.07257501744591766
34	#SeasonOfTheLost	0.07013258897418004
35	#Destiny2Art	0.07013258897418004
36	denied	0.07013258897418004
37	@atmosfioric	0.06978367062107467
38	look	0.06385205861828332
39	was	0.06350314026517795
40	@everyhouranya	0.06245638520586183
41	post	0.06175854849965108
42	appreciation	0.06071179344033496
43	face	0.06001395673412421
44	has	0.058967201674808095
45	@chesscom	0.052686671318911374
46	with	0.04989532449406839
47	ruled	0.04954640614096301
48	over	0.04884856943475227
49	MUST	0.0471039776692254
50	fal	0.04675505931612003
51	Beth	0.044661549197487785
52	Nona	0.03977669225401256
53	Harmon	0.034891835310537335
54	her	0.032449406838799724
55	case	0.0314026517794836
56	can	0.031053733426378228
57	you	0.027564549895324496
58	from	0.025819958129797628
59	line	0.025819958129797628
60	like	0.025471039776692253

	word text	occurrence_rate double precision
61	watch	0.02512212142358688
62	Judge	0.024773203070481507
63	series	0.023028611304954642
64	just	0.0209351011863224
65	@moviemenfes	0.0209351011863224
66	Gaprindashvili	0.020586182833217028
67	Good	0.020237264480111653
68	have	0.019539427773900907
69	Soviet	0.01884159106769016
70	are	0.01884159106769016
71	but	0.018143754361479414
72	Blinders	0.018143754361479414
73	@JenShahade	0.018143754361479414
74	Gambi	0.018143754361479414
75	down	0.018143754361479414
76	Prison	0.01779483600837404
77	man	0.01779483600837404
78	Chess	0.01779483600837404
79	@vibes	0.017445917655268667
80	amp	0.017445917655268667

	word text	occurrence_rate double precision
81	script	0.017096999302163293
82	misrepresented	0.017096999302163293
83	based	0.017096999302163293
84	Break	0.017096999302163293
85	written	0.01674808094905792
86	Surprised	0.01674808094905792
87	boo	0.01674808094905792
88	gambi	0.01674808094905792
89	legendary	0.01674808094905792
90	Picky	0.016399162595952546
91	@Thompson	0.016399162595952546
92	Wife	0.016399162595952546
93	HTGAWM	0.016399162595952546
94	Dexter	0.016399162595952546
95	Mentalist	0.016399162595952546
96	Diaries	0.016399162595952546
97	Castle	0.016399162595952546
98	Blacklist	0.016399162595952546
99	Vampire	0.016399162595952546
100	Mad	0.016399162595952546

b)

```sql
SELECT count(distinct hashtag) AS diff_tag_num
FROM TweetEntity;
```

	diff_tag_num bigint
1	338

c)

```sql
SELECT collection,
    count(hashtag)::float/count(distinct userid)::float as avg_tag_num
FROM(
    SELECT e.tweetid,
        hashtag,
        COALESCE(b.userid, q.userid)                        AS userid,
        CASE WHEN b.tweetid is not null then 'Bridgerton'
            ELSE 'the Queens Gambit'END                     AS collection
    FROM TweetEntity e
    LEFT JOIN BridgertonTweetInfo b
        ON e.tweetid = b.tweetid
    LEFT JOIN QueenTweetInfo q
        ON e.tweetid = q.tweetid
    ) t
GROUP BY collection;
```

	collection text	avg_tag_num double precision
1	Bridgerton	2.58179012345679
2	the Queens Gambit	1.9935064935064934

d)

```sql
SELECT hashtag
FROM(
    SELECT  hashtag,
        count(distinct b.tweetid )    as b_count,
        count(distinct q.tweetid )    as q_count
    FROM TweetEntity e
    LEFT JOIN BridgertonTweetInfo b
        ON e.tweetid = b.tweetid
    LEFT JOIN QueenTweetInfo q
        ON e.tweetid = q.tweetid
    group by hashtag
    ) t
WHERE b_count >0 and q_count>0
```

	hashtag character varying
1	Louies
2	BestFanArmy
3	podcasts
4	Netflix
5	tvtime
6	News
7	BestCoverSong
8	BestMusicVideo
9	Butter
10	HappierThanEver
11	iHeartAwards
12	NEWS
13	bancodeseries
14	ShawnMendes

e)

```sql
SELECT hashtag,b_count,q_count
FROM(
    SELECT  hashtag,
        count(distinct b.tweetid )    as b_count,
        count(distinct q.tweetid )    as q_count
    FROM TweetEntity e
    LEFT JOIN BridgertonTweetInfo b
        ON e.tweetid = b.tweetid
    LEFT JOIN QueenTweetInfo q
        ON e.tweetid = q.tweetid
    group by hashtag
    ) t
WHERE b_count >0 and q_count>0
ORDER BY b_count+q_count DESC
LIMIT 10
```

	hashtag character varying	b_count bigint	q_count bigint
1	Netflix	17	20
2	iHeartAwards	21	5
3	BestFanArmy	14	3
4	tvtime	4	12
5	Louies	9	3
6	HappierThanEver	7	1
7	bancodeseries	1	7
8	podcasts	7	1
9	BestCoverSong	7	1
10	News	2	2

f)

```sql
SELECT b.userid,username,screen_name
FROM BridgertonTweetInfo b
LEFT JOIN StaticUserInfo s
    ON b.userid = s.userid
WHERE b.userid in (
                SELECT userid
                FROM QueenTweetInfo
                )
```

	userid bigint	username character varying	screen_name character varying
1	225187661	GIGICAPONEPR	GiGiCaponePR
2	44686891	Barbie Williams	Barbie892
3	613385915	Lauri Donahue	LauriDonahue
4	14335144	Digital Spy	digitalspy
5	283604227	Andy Vermaut	AndyVermaut
6	1913959446	9 Breaking News	9_breaking_news
7	1257771553988755500	jude | 51 days	jesstfrancis
8	1257771553988755500	jude | 51 days	jesstfrancis
9	171447719	DS Breaking News	digitalspybrk
10	2544099769	🇵🇭▯▯▯▯▯▯ ▯▯▯▯▯▯	falconhamada_90
11	14173315	NBC News	NBCNews
12	577329310	News 100 Media	xedipiworos
13	999907020	OFFICIALGIGICAPONE	msgigicapone
14	280617017	50 Shades of Rein	feeltherein
15	1157981558365667300	Amstel News	amstelnews
16	111556701	Daily Mail Celebrity	DailyMailCeleb
17	456192512	jess	belovedstirling
18	1458943296118612000	Georgina Wiggins	Georginawiggi44
19	456192512	jess	belovedstirling
20	283604227	Andy Vermaut	AndyVermaut
21	1171359256404082700	eyien • DOYOUNG DAY❤️	doyjaefvngrll
22	1434255778462847000	𝑲𝒂𝒕𝒉𝒆𝒓𝒊𝒏𝒆🍺	Katherine2465

	userid bigint	username character varying	screen_name character varying
23	111556701	Daily Mail Celebrity	DailyMailCeleb
24	1561416842	gaby	gabzonfilm
25	280249909	angie =^,^ ia	anjjunie
26	1196845795703742500	Internewscast Media	internewscast
27	4725957312	Radio TFI (Home of The Taxi Stand Hour)	TheRadioTFI
28	1326561978308059100	NewsPlayer+	NewsPlayerPlus
29	821957852	kara aiello	KaraAiello
30	456192512	jess	belovedstirling
31	577329310	News 100 Media	xedipiworos
32	1448332054186299400	chris²³ 🪶	PRdaydreamer
33	225187661	GIGICAPONEPR	GiGiCaponePR
34	1420660507	Woody	Knewz_Currently
35	1313269879	The 206geek	206geek
36	1257771553988755500	jude | 51 days	jesstfrancis
37	225187661	GIGICAPONEPR	GiGiCaponePR
38	1435638352204554200	ReportWire	reportwire_
39	49395291	Charles Myrick	freeacrxcards
40	1257771553988755500	jude | 51 days	jesstfrancis
41	277600246	Gossip Bucket	GossipBucket
42	273353243	Daily News Kit	dailynewskit1
43	4725957312	Radio TFI (Home of The Taxi Stand Hour)	TheRadioTFI
44	456192512	jess	belovedstirling

	userid bigint	username character varying	screen_name character varying
45	787546010	Berkley Bear	BerkleyBearNews
46	1190934402584338400	Kisah kita	Kisahki99454428
47	1221366000752324600	World News	world_news_eng
48	1257771553988755500	jude | 51 days	jesstfrancis
49	4725957312	Radio TFI (Home of The Taxi Stand Hour)	TheRadioTFI
50	1420660507	Woody	Knewz_Currently
51	20578408	Gulf News	gulf_news
52	759655316946554900	Lydia Massiah🌿🍓💖	lydia_massiah
53	2971469687	Rachel	rachelsmith999
54	17434245	The Believer ™	maxvaldes
55	1257771553988755500	jude | 51 days	jesstfrancis
56	1362177019	N A	NahidAlaei
57	1420660507	Woody	Knewz_Currently
58	283604227	Andy Vermaut	AndyVermaut
59	456192512	jess	belovedstirling
60	1257771553988755500	jude | 51 days	jesstfrancis
61	1257771553988755500	jude | 51 days	jesstfrancis
62	999907020	OFFICIALGIGICAPONE	msgigicapone
63	230420580	⚡	rizkafarbilaa
64	1007725923720835100	NewsfeedsMedia	NewsfeedsMedia
65	1898672911	repeatinglitanies	repeatlitanies
66	1293133110876475400	TexasDem	TexasDem2

g)

```sql
SELECT s.userid,
    username,
    screen_name,
    userdescription,
    statuses_count,
    followers_count
FROM StaticUserInfo s
LEFT JOIN DynamicUserInfo d
    ON s.userid = d.userid
ORDER BY d.statuses_count DESC
LIMIT 10;
```

	userid bigint	username character varying	screen_name character varying	userdescription character varying	statuses_count bigint	followers_count bigint
1	289148078	💙 ...*david jones* @MrDJones	MrDJones	You've come to right place. There's NEW...	2582297	5019
2	1420660507	Woody	Knewz_Currently	Love my Family, my Harley, my Jeeps, b...	2551391	1570
3	275942684	rashid al dosari	rashidaldosari	I listen to and respect others views and I...	2272146	2314
4	787546010	Berkley Bear	BerkleyBearNews	Bringing the news that is important, that...	1932470	1382
5	40173650	Global News Report	robinsnewswire	Flying the web for trusted world news re...	1881016	23900
6	1596829382	Zyite.news	ZyiteGadgets	Zyite is a digital media provider	1742740	3216
7	49404200	Charles Myrick -CEO	medicinehelp	American Consultants Rx is focused on ...	1696332	1358
8	2544099769	🇪🇬▯▯▯▯▯▯▯▯ ▯▯▯▯▯▯▯	falconhamada_90	Official Account	1629003	2343
9	1221366000752324600	World News	world_news_eng	For All English Users	1590967	2444
10	3534222021	NewsOnePlace.com	newsoneplace	https://t.co/WWMmlkl7DB is a news ag...	1586895	998

h)

```sql
CREATE INDEX gin_text_b ON BridgertonTweetInfo USING gin(to_tsvector('english', text));
CREATE INDEX gin_text_q ON QueenTweetInfo USING gin(to_tsvector('english', text));
```

i)

Finding：

I filter the tweets about Brigerton that have been liked by at least 100 Twitter users. And I choose my keyword by looking at the most frequent words in those “liked” tweets. I found people are talking about the new season of Brigerton which will come in Feburary and the final ball in Brigerton. About 30% tweets that I collected are taking about this topic.

```sql
--Find the keywords using favorite count
SELECT lower(word),count(1) as word_count
FROM  BridgertonTweetInfo,UNNEST(regexp_split_to_array(
                left(text,position('https://'in text)-1) --ignore the urls in tweet
                ,'[^A-Za-z0-9@]') -- split by special characters
           ) AS word
WHERE (raw_tweet->'favorite_count'):: text::int >= 100
    AND length(word)>=3 --ingore word shorter than 3 characters
GROUP BY lower(word)
ORDER BY word_count desc

--Search with keywords (february,month,ball,final,coming,next,season)
SELECT tweetid,text
FROM  BridgertonTweetInfo
WHERE lower(text) SIMILAR TO '%(february|month|ball|final|next|season|coming)%'

--Check the porpotion of tweets with keywords
SELECT count(k.tweetid) as ktweets_num,
    count(b.tweetid)    as total_tweets_num,
    count(k.tweetid)::float/count(b.tweetid)::float as rate
FROM BridgertonTweetInfo b
LEFT JOIN (
    SELECT tweetid,text
    FROM  BridgertonTweetInfo
    WHERE lower(text) SIMILAR TO '%(february|month|ball|final|next|season|coming)%'
    ) k
    ON b.tweetid = k.tweetid
```

	lower text	word_count bigint
1	bridgerton	25
2	the	12
3	february	4
4	and	4
5	month	4
6	ball	4
7	she	4
8	her	4
9	his	3
10	kate	3
11	final	3
12	was	3
13	next	3
14	star	3
15	can	3
16	out	3
17	season	3
18	happy	2
19	relaxed	2
20	coming	2
21	confirming	2

	tweetid bigint	text text
1	1488721839887384600	RT @Anladybee: Kate Sharma in Bridgerton Season 2 https://t.co/PtBElleGbT
2	1488505691346436000	RT @chaoticguitar: FINAL BALL OF BRIDGERTON S2 COULD BE A MASQUERADE BALL IM FREAKING OUT CHRIS VAN DUSEN SKSKSKSKSKSS https://t.co/wkmAbaJ…
3	1487091052821446700	RT @KennethDredd: Bill Ackman 2020: "Hell is coming"
4	1488505741094781000	RT @chaoticguitar: FINAL BALL OF BRIDGERTON S2 COULD BE A MASQUERADE BALL IM FREAKING OUT CHRIS VAN DUSEN SKSKSKSKSKSS https://t.co/wkmAbaJ…
5	1488367921675993000	RT @BuzzFeed: Bridgerton star Nicola Coughlan would kindly appreciate it if you kept your opinions on her body to yourself. https://t.co/fW…
6	1488607929825640400	RT @NeenaQueen20: February 1. That means we get Bridgerton next month. We get to see the most anticipated book of the series happen NEXT MO…
7	1488546535721869300	RT @chaoticguitar: Good morning! Happy February and 52 days to #Bridgerton 🐝✨💞 The month of love is upon us weeeee! https://t.co/nJxGmETaMK
8	1488512557598052400	"I'm not a body positivity activist, I'm an actor."
9	1487483267041398800	RT @kathonysbee: No idea if anybody here cares about this modern Kathony AU, but chapter 3 of 'Tis The Damn Season is up! No smut in this o…
10	1488440672025776000	i finished bridgerton now i'm waiting on season 2 🤦 iykyk
11	1487343238281314300	RT @Anladybee: Kate Sharma in Bridgerton Season 2 https://t.co/PtBElleGbT
12	1487170303948165000	@brauhala hear me out…for your next "Broadway Loves" series…The (Unofficial) Bridgerton Musical by @abigailbarloww… https://t.co/h1tjkx2DdM
13	1487285813872365600	Bridgerton season 2 when
14	1487169778720641000	RT @menacetoanthony: Can't wait to see more of Shelley Conn as Mary Sharma in season 2 of Bridgerton! Look how lovely she is 💗☺
15	1487022402588577800	Nicola Coughlan teases steamier scenes between Penelope and Colin in #Bridgerton season 2:
16	1488169675452260400	RT @NoContxtBR: 53 days until we finally see the real star of #bridgerton https://t.co/ZtrHseq4zl
17	1488560141238734800	that's great bridgerton teaser next tho? https://t.co/ftZgptOjW8
18	1488387433301299200	RT @BuzzFeed: Bridgerton star Nicola Coughlan would kindly appreciate it if you kept your opinions on her body to yourself. https://t.co/fW…
19	1488444323779727400	RT @NoContxtBR: 53 days until we finally see the real star of #bridgerton https://t.co/ZtrHseq4zl
20	1488250641424261000	RT @CNN: 'Bridgerton' star Nicola Coughlan has asked people to stop commenting on her body -- the latest celeb to speak out about bodyshami…
21	1487052284043350000	I've starting watching Bridgerton, and watching Simon tell Daphne that they were never really friends is just watch… https://t.co/AKoOUHJOoq
22	1487445515478388700	RT @Anladybee: Kate Sharma in Bridgerton Season 2 https://t.co/PtBElleGbT

	ktweets_num bigint	total_tweets_num bigint	rate double precision
1	1593	5009	0.31802755040926334