

# Baseball Database

Jing Wang/Siwei Zeng

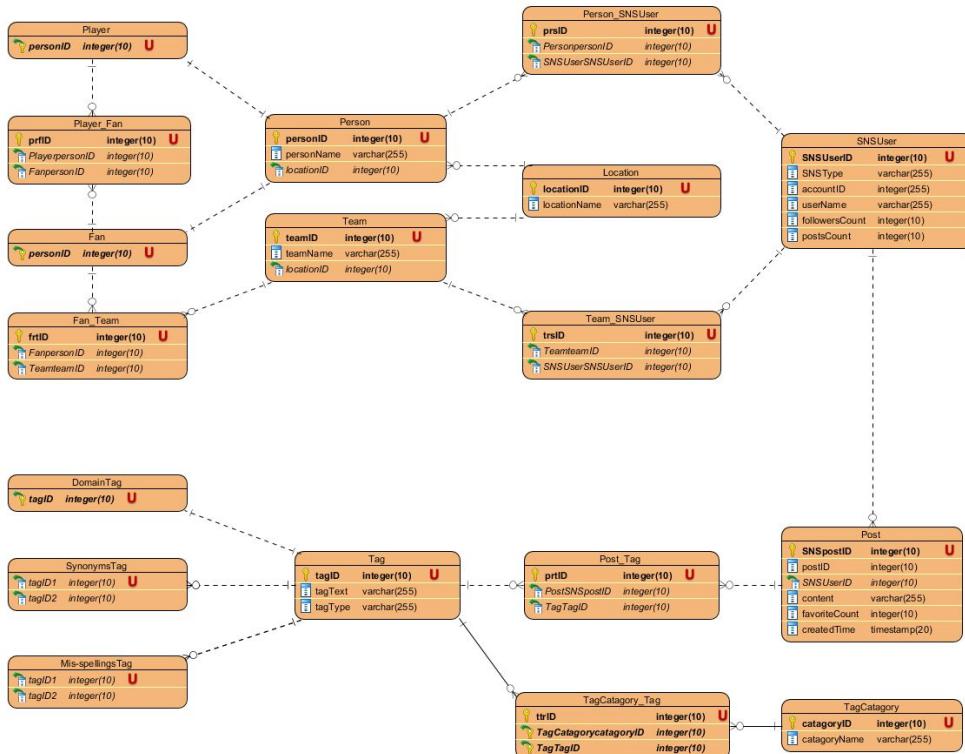
## Background

In this project, we are assumed to be working for a company called Nerd Analytics. Our job is creating a social news site. We choose to build a database about baseball. Each domain must have entities that represent people, places and things. In our project, baseball player and fan represent people, team represents thing and location represents place. The whole project includes building conceptual model, acquiring data from Twitter and Instagram, Tagging and writing use case according to our database.

## Procedure Diagram



# Entity Relationship Diagram



## Data Acquisition and Tagging

We get the data from Instagram and Twitter by Instaloader and Tweepy. For the convenience, we define many functions first to use the API to get the data. We use hashtag and mention to do the data tagging. Then we convert these tables to csv files.

### 1) Instagram:

#### import libraries and setting instaloader

```
In [1]: import instaloader
import datetime
import warnings
import time
warnings.filterwarnings("ignore")
import time
from instaloader import Instaloader,Profile
L=Instaloader()
```

## define API functions

```
] def anyname2user(anyname):
    if type(anyname)==int:
        profile=Profile.from_id(L.context,anyname)
    else:
        profile=Profile.from_username(L.context,anyname)
    return profile
```

## Make DataFrame

```
import pandas as pd
Player = pd.DataFrame(columns=["personID"])
Player_Fan = pd.DataFrame(columns=["prfID", "PlayerpersonID", "FanpersonID"])
Fan = pd.DataFrame(columns=["personID"])
Fan_Team = pd.DataFrame(columns=["frtID", "FanpersonID", "TeamteamID"])
Person = pd.DataFrame(columns=["personID", "personName", "locationID"])
Team = pd.DataFrame(columns=["teamID", "teamName", "locationID"])
Person_SNSUser = pd.DataFrame(columns=["prsID", "PersonpersonID", "SNSUserSNSUserID"])
Location = pd.DataFrame(columns=["locationID", "locationName"])
Team_SNSUser = pd.DataFrame(columns=["trsID", "TeamteamID", "SNSUserSNSUserID"])
SNSUser = pd.DataFrame(columns=["SNSUserID", "SNSType", "accountID", "userName", "followersCount", "postsCount"])
Post = pd.DataFrame(columns=["SNSpostID", "postID", "SNSUserID", "content", "favoriteCount", "createdTime"])
Post_Tag = pd.DataFrame(columns=["prtID", "PostSNSpostID", "TagTagID"])
Tag = pd.DataFrame(columns=["tagID", "tagText", "tagType"])
DomainTag = pd.DataFrame(columns=["tagID"])
SynonymsTag = pd.DataFrame(columns=["tagID1", "tagID2"])
MisspellingsTag = pd.DataFrame(columns=["tagID1", "tagID2"])
TagCatagory_Tag = pd.DataFrame(columns=["ttrID", "TagCatagorycatagoryID", "TagTagID"])
TagCatagory = pd.DataFrame(columns=["catagoryID", "catagoryName"])
```

## Data Retrieving Function

```
def savesnsuser(user):
    if SNSUser[SNSUser["accountID"].isin([user2accountid(user)])].shape[0] == 0:
        SNSUser.loc[SNSUser.shape[0] + 1] = {"SNSUserID": SNSUser.shape[0] + 1,
                                              "SNSType": "Twitter",
                                              "accountID": user2accountid(user),
                                              "userName": user2username(user),
                                              "followersCount": user2followerscount(user),
                                              "postsCount": user2postscount(user)}
    else:
        SNSUser.set_value(SNSUser[SNSUser["accountID"].isin([user2accountid(user)])].loc[:, ["SNSUserID"]], SNSUser[SNSUser["accountID"].isin([user2accountid(user)])].loc[:, ["SNSUserID"]])
```

### domain tag

```
teamSNS=[["New York Yankees", "yankees"],
         ["Los Angeles Dodgers", "dodgers"],
         ["Tampa Bay Rays", "raysbaseball"]]
playerSNS=[["Greg Bird", "_gregbird33"],
           ["DJ LehMieeu", "dj_lehmieeu"],
           ["Zack Britton", "zbritton"],
           ["Max Muncy", "maxmuncy"],
           ["Julio Urias", "theteenager7"],
           ["Cody Bellinger", "cody_bellinger"],
           ["Christian Arroyo", "arroyoc22"],
           ["Austin Meadows", "austinmeadows13"],
           ["Guillermo Heredia", "heredia54"]]

for tag in Taglist:
    if tag[2]=="hashtag":
        for player in playerSNS:
            if tag[1] in player[1]:
                DomainTagList.append([tag[0]])
        for team in teamSNS:
            if tag[1] in team[1]:
                DomainTagList.append([tag[0]])
```

### synonyms tag

```
def tagcount1(tagid):
    counter = 0
    for pt in Post_TagList:
        if pt[2]==tagid:
            counter=counter+1
    return counter
tagcount1(5)
```

## Misspelling Tag

```
# Levenshtein distance calculation function
# Reference: https://rosettacode.org/wiki/Levenshtein_distance#Python
def ldistance(s1, s2):
    if len(s1) > len(s2):
        s1, s2 = s2, s1
    distances = range(len(s1) + 1)
    for index2, char2 in enumerate(s2):
        newDistances = [index2 + 1]
        for index1, char1 in enumerate(s1):
            if char1 == char2:
                newDistances.append(distances[index1])
            else:
                newDistances.append(1 + min((distances[index1],
                                              distances[index1 + 1],
                                              newDistances[-1])))
        distances = newDistances
    return distances[-1]
```

## Tagging dataframe

```
: def list2tuplelist(list):
    tuplelist=[]
    for row in list:
        tuplelist.append(tuple(row))
    return tuplelist
```

2) Twitter:

### Import Libraries and Setting API

```
1 import tweepy
2 import json
3 import datetime
4 import warnings
5 import time
6 warnings.filterwarnings('ignore')
7 consumer_key = 'nR08b2R7tnipKtrFTJVRUru7V'
8 consumer_secret = 'bK2v3PVVZGktAT2BviC96Tz9vzjq0ZDj2WLKZ1pmm5aGgu4eW'
9 access_token = '1098973181468205057-xFZG8xbtYbXjE6T2s00k5J0bKRpsN'
10 access_token_secret = 'W7mu0M4ABrbSH59nHLH43YMeRl3GZcfN1v8FOh1p8tPHh'
11 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
12 auth.set_access_token(access_token, access_token_secret)
13 api = tweepy.API(auth, wait_on_rate_limit = True)
```

## Make DataFrame

```
1 import pandas as pd
2 Player = pd.DataFrame(columns = ["personID"])
3 Player_Fan = pd.DataFrame(columns = ["prfID", "PlayerpersonID", "FanpersonID"])
4 Fan = pd.DataFrame(columns = ["personID"])
5 Fan_Team = pd.DataFrame(columns = ["frtID", "FanpersonID", "TeamteamID"])
6 Person = pd.DataFrame(columns = ["personID", "personName", "locationID"])
7 Team = pd.DataFrame(columns = ["teamID", "teamName", "locationID"])
8 Person_SNSUser = pd.DataFrame(columns = ["prsID", "PersonpersonID", "SNSUserSNSUserID"])
9 Location = pd.DataFrame(columns = ["locationID", "locationName"])
10 Team_SNSUser = pd.DataFrame(columns = ["trsID", "TeamteamID", "SNSUserID"])
11 SNSUser = pd.DataFrame(columns = ["SNSUserID", "SNSType", "accountID", "userName", "followersCount", "postsCount"])
12 Post = pd.DataFrame(columns = ["SNSpostID", "postID", "SNSUserID", "content", "favoriteCount", "createdTime"])
13 Post_Tag = pd.DataFrame(columns = ["prtID", "PostSNSpostID", "TagTagID"])
14 Tag = pd.DataFrame(columns = ["tagID", "tagText", "tagType"])
15 DomainTag = pd.DataFrame(columns = ["tagID"])
16 SynonymsTag = pd.DataFrame(columns = ["tagID1", "tagID2"])
17 MisspellingsTag = pd.DataFrame(columns = ["tagID1", "tagID2"])
18 TagCatagory_Tag = pd.DataFrame(columns = ["trtID", "TagCatagorycatagoryID", "TagTagID"])
19 TagCatagory = pd.DataFrame(columns = ["catagoryID", "catagoryName"])
```

## DomainTag

```
1 for tag in TagList:
2     if tag[2]=="hashtag":
3         for player in playerSNS:
4             if tag[1] in player[1]:
5                 DomainTagList.append([tag[0]])
6         for team in teamSNS:
7             if tag[1] in team[1]:
8                 DomainTagList.append([tag[0]])
9 DomainTagList
```

## SynonymsTag

```
1 ratioleast = 0.75
2 countleast = 10
3 for tag1 in TagList:
4     if tag1[2] == "hashtag" and tagcount1(tag1[0])>countleast:
5         for tag2 in TagList:
6             if tag2[2] == "hashtag" and tagcount1(tag2[0])>countleast:
7                 if tag1 != tag2:
8                     tag1id = tag1[0]
9                     tag2id = tag2[0]
10                    f1 = tagcount1(tag1id)
11                    f2 = tagcount1(tag2id)
12                    f3 = tagcount2(tag1id, tag2id)
13                    if f3/f1 > ratioleast or f3/f2 >ratioleast:
14                        if tag1id > tag2id:
15                            temptagid = tag1id
16                            tag1id = tag2id
17                            tag2id = temptagid
18                        if [tag1id, tag2id] not in SynonymsTagList:
19                            SynonymsTagList.append([tag1id, tag2id])
```

## Levenshtein distance calculation function

```
1 # Levenshtein distance calculation function
2 def ldistance(s1, s2):
3     if len(s1) > len(s2):
4         s1, s2 = s2, s1
5     distances = range(len(s1) + 1)
6     for index2, char2 in enumerate(s2):
7         newDistances = [index2 + 1]
8         for index1, char1 in enumerate(s1):
9             if char1 == char2:
10                 newDistances.append(distances[index1])
11             else:
12                 newDistances.append(1 + min((distances[index1],
13                                     distances[index1 + 1],
14                                     newDistances[-1])))
15         distances = newDistances
16     return distances[-1]
```

## Tagging dataframe

```
1 def list2tuplelist(list):
2     tuplelist=[]
3     for row in list:
4         tuplelist.append(tuple(row))
5     return tuplelist
1 labels=["tagID"]
2 DomainTag=pd.DataFrame.from_records(list2tuplelist(DomainTagList), columns=labels)
```

### 3) Sample data:

```
1 Player.head()
```

personID	
1	1
2	2
3	3
4	4
5	5

```
1 Player_Fan.head()
```

	prflID	PlayerpersonID	FanpersonID
1	1	3	684
2	2	5	758
3	3	5	759
4	4	5	760
5	5	5	761

```
1 | Fan.head()
```

personID
1
2
3
4
5

```
1 | Fan_Team.head()
```

frtID	FanpersonID	TeamteamID
1	1	10
2	2	11
3	3	12
4	4	13
5	5	14

```
1 | Person.head()
```

personID	personName	locationID
1	Alexander David Bregman	4
2	Carlos Javier Correa	5
3	José Carlos Altuve	5
4	Nicholas A. Castellanos	6
5	Jose Miguel Cabrera	7

```
1 | Team.head()
```

teamID	teamName	locationID
1	Houston Astros	1
2	Detroit Tigers	2
3	Texas Rangers	3

```
1 | Person_SNSUser.head()
```

prsID	PersonpersonID	SNSUserSNSUserID
1	1	4
2	2	5
3	3	6
4	4	7
5	5	8

```
1 | Location.head()
```

locationID	locationName
1	Minute Maid Park
2	Comerica Park
3	Deep in the Heart of Texas
4	Albuquerque, NM
5	

```
1 | Team_SNSUser.head()
```

trsID	TeamteamID	SNSUserSNSUserID
1	1	1
2	2	2
3	3	3

```
1 | SNSUser.head()
```

SNSUserID	SNSType	accountID	userName	followersCount	postsCount
1	1	Twitter	52803520	astros	1455653
2	2	Twitter	30008146	tigers	1453071
3	3	Twitter	40931019	Rangers	1426982
4	4	Twitter	382708289	ABREG_1	242213
5	5	Twitter	2811330143	TeamCJCorrea	221551

```
1 | Post.head()
```

SNSpostID	postID	SNSUserID	content	favoriteCount	createdTime
1	1	1121607683507810304	Final: Indians 2, #Astros 1	111	2019-04-26 02:51:24
2	2	1121598945145147392	We know you'd like this in your living room. \...	1068	2019-04-26 02:16:41
3	3	1121582532955230208	Hang it, then bang it. 🎯\n\nWe're all square! ...	2829	2019-04-26 01:11:28
4	4	1121579802551435264	George doesn't look too frustrated anymore.\n\...	1453	2019-04-26 01:00:37
5	5	1121575278755471360	Special message from the @Indians to Uncle Mik...	623	2019-04-26 00:42:38

```
1 Post_Tag.head()
```

prtID	PostSN	postID	TagTagID
1	1	1	1
2	2	2	2
3	3	3	2
4	4	4	2
5	5	5	2

```
1 Tag.head()
```

tagID	tagText	tagType
1	Astros	hashtag
2	TakeItBack	hashtag
3	Indians	mention
4	ABREG_1	mention
5	ATTsportsNetSW	mention

```
1 DomainTag.head()
```

tagID
0
2553
1
2571
2
2571
3
2613
4
2656

```
1 SynonymsTag.head()
```

tagID1	tagID2
0	1
1	55
1	1
102	
2	1
2649	
3	1
2650	
4	1
2651	

```
1 MisspellingsTag.head()
```

tagID1	tagID2
0	37
1	1
2613	
2	1
2689	
3	16
16	2
4	2
2626	

```
1 TagCatagory_Tag.head()
```

	trtID	TagCatagorycatagoryID	TagTagID
0	1	1	1
1	2	1	2
2	3	2	3
3	4	2	4
4	5	2	5

```
1 TagCatagory.head()
```

	catagoryID	catagoryName
0	1	hashtag
1	2	mention
2	3	domaintag
3	4	synonymstag
4	5	misspellingstag

## Check the Table

- 1) 1NF check: Each table has a primary key; The values in each column of a table are atomic; There are no repeating groups.
- 2) 2 NF check: No partial dependencies; No calculated data.
- 3) 3NF check: Eliminate fields that do not directly depend on the primary key, that is no transitive dependencies.

## Newsfeed (popular hashtags)

- 1) What are people saying about me (somebody)?

```
1 SELECT * FROM post WHERE SNSpostID IN (
2   SELECT PostSNSpostID FROM post_tag WHERE TagTagID IN (
3     SELECT tagID FROM tag WHERE tagText = "ABREG_1" AND tagType = "mention"))
```

The screenshot shows a database result grid with the following columns: SNSpostID, postID, SNSUserID, content, favoriteCount, and createdTime. The data includes various tweets from users like @ABREG\_1, @Clutch, and @Astros, with their respective IDs and creation times.

SNSpostID	postID	SNSUserID	content	favoriteCount	createdTime
7	1121560073720008704	1	To close out Autism awareness month, @ABRE...	933	2019-04-25 23:42:13
47	1120868046883528704	1	Clutch. @ABREG_1 drives home 2 to take the l...	2145	2019-04-24 01:52:21
67	1120508481453166592	1	Who's smoother than @ABREG_1? #TakeItBack...	877	2019-04-23 02:03:34
108	1119703315397345282	1	Unofficial #Astros marketing spokesperson, @A...	3170	2019-04-20 20:44:08
158	111789268432576513	1	RT @ABREG_1: Thank you Jada! <a href="https://t.co/...">https://t.co/...</a> 0	0	2019-04-15 20:47:40
325	1114610992367972352	1	RT @ABREG_1: Wish you were here @trviaXX.h...	0	2019-04-06 19:29:03
341	1114335663388680193	1	First RBI of the home slate. Courtesy: @ABRE...	836	2019-04-06 01:15:00
427	1112050016540086273	1	RT @InfieldChatter: .@astros 3B Alex Bregman...	0	2019-03-30 17:52:39
429	1112006272226869249	1	Huuuuuge day celebrating @ABREG_1 and @35...	9664	2019-03-30 14:58:50

## 2) How viral are my posts?

```
1   SELECT AVG(favoriteCount) FROM post WHERE SNSUserID IN(  
2     SELECT SNSUserID FROM snsuser WHERE userName = "astros")
```

The screenshot shows a database query results grid. The top bar includes 'Result Grid', 'Filter Rows:', 'Export:', and 'Wrap Cell Content'. On the right, there are three icons: 'Result Grid' (selected), 'Form Editor', and 'Field Types'. The results table has one row with the following data:

	AVG(favoriteCount)
	1540.7459

## 3) What posts are likely to be interesting to me?

```
1 • ⚡ SELECT * FROM post WHERE SNSpostID IN(  
2   ⚡ SELECT DISTINCT PostSNSpostID FROM post_tag WHERE TagTagID IN(  
3     ⚡ SELECT DISTINCT TagTagID FROM post_tag WHERE PostSNSpostID IN(  
4       ⚡ SELECT SNSpostID FROM post WHERE SNSUserID IN(  
5         ⚡ SELECT SNSUserID FROM snsuser WHERE userName = "astros"))))
```

The screenshot shows a database query results grid. The top bar includes 'Result Grid', 'Filter Rows:', 'Export:', and 'Wrap Cell Content'. On the right, there are three icons: 'Result Grid' (selected), 'Form Editor', and 'Field Types'. The results table has 8 rows of data:

SNSpostID	postID	SNSUserID	content	favoriteCount	createdTime
1	1121607683507810304	1	Final: Indians 2, #Astros 1	111	2019-04-26 02:51:24
2	1121598945145147392	1	We know you'd like this in your living room. #T...	1068	2019-04-26 02:16:41
3	1121582532955230208	1	Hang it, then bang it. We're all square! #T...	2829	2019-04-26 01:11:28
4	1121579802551435264	1	George doesn't look too frustrated anymore. #T...	1453	2019-04-26 01:00:37
5	1121575278755471360	1	Special message from the @Indians to Uncle Mik...	623	2019-04-26 00:42:38
6	1121567325096800256	1	We couldn't ask for a more perfect night. #Ta...	658	2019-04-26 00:11:02
7	1121560073720008704	1	To close out Autism awareness month, @ABRE...	933	2019-04-25 23:42:13
8	1121537600077454154	1	Thursday showdown 🏈 OPEN 5:00pm 7...	400	2019-04-25 21:53:03

Post 37 x Read Only

#### 4) What posts are like mine?

```

1 • Ⓜ SELECT * FROM post WHERE SNSpostID IN(
2 Ⓜ SELECT DISTINCT PostSNSpostID FROM post_tag WHERE TagTagID IN(
3 Ⓜ SELECT DISTINCT TagTagID FROM post_tag WHERE PostSNSpostID IN(
4 Ⓜ SELECT SNSpostID FROM post WHERE SNSUserID IN(
5   - SELECT SNSUserID FROM snsuser WHERE userName = "astros"))
6 Ⓜ AND PostSNSpostID NOT IN(
7   - SELECT SNSpostID FROM post WHERE SNSUserID IN(
8     - SELECT SNSUserID FROM snsuser WHERE userName = "astros"))

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

SNSpostID	postID	SNSUserID	content	favoriteCount	createdTime
668	1120879397039411200	2	@ltwhale @mams33 @DetroitRedWings @Lions ...	9	2019-04-24 02:37:27
763	112083350200119297	2	@woody_ @MLB This is an elite tweet	1	2019-04-23 23:34:29
823	1120684273424580615	2	@espn @ us next time https://t.co/BuewTHwMH2	2767	2019-04-23 13:42:06
899	1120048092013891585	2	@Indians You're punnier than this, AtIndians.	26	2019-04-21 19:34:09
1110	111898411889052672	2	@MLB https://t.co/Qx7tWVL	380	2019-04-18 21:06:23
1113	1118983138120732676	2	RT @MLB: It's Miggy's birthday, so let's take a l...	0	2019-04-18 21:02:24
1204	1118900597548355585	2	@Indians Luv u, mean it.	37	2019-04-18 15:34:25
1240	1118607123358978054	2	RT @MLB: Ten tit #MTFino? No, 16 overall pro...	0	2019-04-18 01:46:01

#### 5) What users post like me?

```

1 • Ⓜ SELECT DISTINCT SNSUserID FROM post WHERE SNSpostID IN(
2 Ⓜ SELECT DISTINCT PostSNSpostID FROM post_tag WHERE TagTagID IN(
3 Ⓜ SELECT DISTINCT TagTagID FROM post_tag WHERE PostSNSpostID IN(
4 Ⓜ SELECT SNSpostID FROM post WHERE SNSUserID IN(
5   - SELECT SNSUserID FROM snsuser WHERE userName = "astros"))
6 Ⓜ AND PostSNSpostID NOT IN(
7   - SELECT SNSpostID FROM post WHERE SNSUserID IN(
8     - SELECT SNSUserID FROM snsuser WHERE userName = "astros"))

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

SNSUserID
2

#### 6) Who should I be following?

```

1 • Ⓜ SELECT * FROM snsuser WHERE SNSUserID IN (
2 Ⓜ SELECT DISTINCT SNSUserID FROM post WHERE SNSpostID IN(
3 Ⓜ SELECT DISTINCT PostSNSpostID FROM post_tag WHERE TagTagID IN(
4 Ⓜ SELECT DISTINCT TagTagID FROM post_tag WHERE PostSNSpostID IN(
5 Ⓜ SELECT SNSpostID FROM post WHERE SNSUserID IN(
6   - SELECT SNSUserID FROM snsuser WHERE userName = "astros"))
7 Ⓜ AND PostSNSpostID NOT IN(
8   - SELECT SNSpostID FROM post WHERE SNSUserID IN(
9     - SELECT SNSUserID FROM snsuser WHERE userName = "astros"))))

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

SNSUserID	SNSType	accountID	userName	followersCount	postsCount
2	Twitter	30008146	tigers	1453071	49251

## 7) What topics are trending in my domain?

```
1 • SELECT DISTINCT TagTagID, COUNT(TagTagID) FROM post_tag WHERE PostSNSpostID IN(
2   SELECT SNSpostID FROM post WHERE SNSUserID IN(
3     SELECT SNSUserID FROM snsuser WHERE userName = "astros"))
4   ORDER BY COUNT(TagTagID)
5   LIMIT 10
```

Result Grid	
TagTagID	COUNT(TagTagID)
1	688

## 8) What keywords/ hashtags should I add to my post?

- ```
SELECT DISTINCT TagTagID, ztag.tagText, SUM(zpost.favoriteCount) AS favoriteCount
  FROM zpost_tag
  INNER JOIN zpost
  ON zpost_tag.PostSNSpostID = zpost.SNSpostID
  INNER JOIN ztag
  ON zpost_tag.TagTagID = ztag.tagID
  GROUP BY zpost_tag.TagTagID
  ORDER BY favoriteCount DESC
  LIMIT 5
```

## 9) Should I follow somebody back?

```
1 • SELECT * FROM zsnsuser
2   WHERE accountID IN (SELECT DISTINCT SNSUserID FROM zpost
3   WHERE SNSpostID IN (SELECT DISTINCT PostSNSpostID FROM zpost_tag
4   WHERE TagTagID IN (SELECT DISTINCT TagTagID FROM zpost_tag
5   WHERE PostSNSpostID IN (SELECT SNSpostID FROM zpost
6   WHERE SNSUserID IN (SELECT SNSUserID FROM zsnsuser
7   WHERE userName = "''' + ''' + me + ''' + '''))))
```

# 10 Use Cases

1.Which ins post is the most popular?

```
1 •  select * from wpost order by favoriteCount DESC limit 1
```

A screenshot of a database query results grid. The top bar shows "100%" and "48:1". The toolbar includes "Result Grid", "Filter Rows", "Search", "Export", and "Fetch rows". The table has columns: SNSpo..., postID, SNSUs..., content, favoriteCo..., and createdTime. A single row is shown: 9, 2.03e18, 2, Ya baby!!! @dugie11 ??..., 71886, 2019/4/25 19:49.

| SNSpo... | postID  | SNSUs... | content                   | favoriteCo... | createdTime     |
|----------|---------|----------|---------------------------|---------------|-----------------|
| 9        | 2.03e18 | 2        | Ya baby!!! @dugie11 ??... | 71886         | 2019/4/25 19:49 |

2.Which ins post is the least popular?

```
1 •  select * from wpost order by favoriteCount limit 1
```

A screenshot of a database query results grid. The top bar shows "100%" and "43:1". The toolbar includes "Result Grid", "Filter Rows", "Search", "Export", and "Fetch rows". The table has columns: SNSpo..., postID, SNSUs..., cont..., favoriteCo..., and createdTime. A single row is shown: 13, 2.03e18, 13, 2, 2019/4/26 10:59.

| SNSpo... | postID  | SNSUs... | cont... | favoriteCo...   | createdTime |
|----------|---------|----------|---------|-----------------|-------------|
| 13       | 2.03e18 | 13       | 2       | 2019/4/26 10:59 |             |

3.Which twitter post is the most popular?

```
1 •  select * from zpost order by favoriteCount DESC limit 1
```

The screenshot shows a MySQL Workbench interface with a query editor at the top containing the SQL code. Below it is a "Result Grid" window displaying the results of the query. The results show one row of data:

| SNSpo... | postID              | SNSUs... | content      | favoriteCo... | createdTime          |
|----------|---------------------|----------|--------------|---------------|----------------------|
| 317      | 1114681703664996... | 1        | La Flame 🎉🔥🎉 | 15302         | 2019-04-07 00:10:... |

4.Which twitter post is the least popular?

```
1 •  select * from zpost order by favoriteCount limit 1
```

The screenshot shows a MySQL Workbench interface with a query editor at the top containing the SQL code. Below it is a "Result Grid" window displaying the results of the query. The results show one row of data:

| SNSpo... | postID              | SNSUs... | content                                      | favoriteCo... | createdTime |
|----------|---------------------|----------|----------------------------------------------|---------------|-------------|
| 12       | 1121462699408465... | 1        | RT @RealJoshReddick: Met the Monster Amon... | 0             | 2019-04-25  |

5. How many favorites did the latest twitter post get?

```
1 •  select postID, content, favoriteCount from zpost  
2      order by createdTime limit 1
```

A screenshot of a database query results grid. At the top, there are zoom controls (100%, 1:2) and a toolbar with icons for Result Grid, Filter Rows, Search, Export, and Fetch rows. The result grid has three columns: postID, content, and favoriteCo... (truncated). One row is visible, showing postID 1110874191790829..., content "Tomorrow.", and favoriteCount 1511.

| postID                | content   | favoriteCo... |
|-----------------------|-----------|---------------|
| ► 1110874191790829... | Tomorrow. | 1511          |
|                       |           |               |

6. How many favorites did the latest ins post get?

```
1 •  select postID, content, favoriteCount from wpost  
2      order by createdTime limit 1
```

A screenshot of a database query results grid. At the top, there are zoom controls (100%, 29:2) and a toolbar with icons for Result Grid, Filter Rows, Search, Export, and Fetch. The result grid has three columns: postID, cont... (truncated), and favoriteCo... (truncated). One row is visible, showing postID 1.63e18 and favoriteCount 12.

| postID    | cont... | favoriteCo... |
|-----------|---------|---------------|
| ► 1.63e18 | 12      |               |
|           |         |               |

7. Rank the ins posts according to the favorite count.

```
1 •  select * from wpost order by favoriteCount
```

The screenshot shows a database query results grid. At the top, there is a command line interface with the text "1 • select \* from wpost order by favoriteCount". Below this is a toolbar with "Result Grid", "Filter Rows", "Search", and "Export" buttons. The main area displays a table with the following data:

| SNSpo... | postID  | SNSUs... | content                                              | favoriteCo... | createdTim... |
|----------|---------|----------|------------------------------------------------------|---------------|---------------|
| 11       | 2.03e18 | 11       | Nicoooooo.                                           | 3             | 2019/4/25 1   |
| 16       | 1.63e18 | 16       |                                                      | 12            | 2017/10/18    |
| 15       | 2.03e18 | 15       |                                                      | 48            | 2019/4/26 6   |
| 11       | 2.03e18 | 3        | Holey biceps! ??                                     | 4975          | 2019/4/25 1   |
| 2        | 2.03e18 | 2        | Amazing pic????                                      | 12088         | 2019/4/25 2   |
| 1        | 2.03e18 | 2        | Can we a do a friends night ???                      | 12119         | 2019/4/26 0   |
| 4        | 2.03e18 | 2        | Bullpen was solid... they deserve credit, they we... | 23631         | 2019/4/25 2   |
| 10       | 2.03e18 | 2        | I listening at work @dnndnrs ??                      | 24277         | 2019/4/25 1   |

8. Rank the twitter posts according to the favorite count.

```
1 •  select * from zpost order by favoriteCount
```

The screenshot shows a database query results grid. At the top, there is a command line interface with the text "1 • select \* from zpost order by favoriteCount". Below this is a toolbar with "Result Grid", "Filter Rows", "Search", "Export", and "Fetch rows" buttons. The main area displays a table with the following data:

| SNSpo... | postID              | SNSUs... | content                                         | favoriteC... |
|----------|---------------------|----------|-------------------------------------------------|--------------|
| 12       | 1121462699408465... | 1        | RT @RealJoshReddick: Met the Monster Amon...    | 0            |
| 29       | 1121143886779494... | 1        | RT @ATTSportsNetSW: Tonight, @astros vs Tw...   | 0            |
| 32       | 1121123419448332... | 1        | RT @BraunStrowman: Pre first pitch pump eng...  | 0            |
| 33       | 1121105565911060... | 1        | RT @MLB: These infielders are crushing.         | 0            |
| 40       | 1120890826945511... | 1        | RT @AstrosTrainGuy: You know who can hit a b... | 0            |
| 59       | 1120732323874471... | 1        | RT @HouOpenGolf: Updates to the 19th Hole!      | 0            |
| 98       | 1119970808389148... | 1        | RT @RealJoshReddick: Happy Easter everyon...    | 0            |

## 9. What posts did people mention about ABREG\_1?

```
1  SELECT * FROM post WHERE SNSpostID IN (
2      SELECT PostSNSpostID FROM post_tag WHERE TagTagID IN (
3          SELECT tagID FROM tag WHERE tagText = "ABREG_1" AND tagType = "mention"))
```

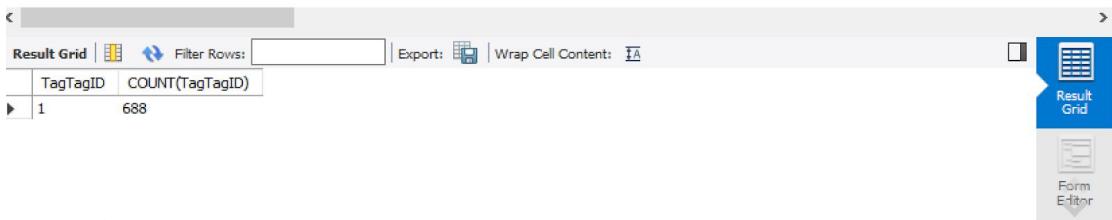


The screenshot shows a database query results grid titled "Result Grid". The grid displays 10 rows of data from a table with columns: SNSpostID, postID, SNSUserID, content, favoriteCount, and createdTime. The content column shows various tweets from users like @ABREG\_1, @Clutch, and @Astros. The favoriteCount and createdTime columns provide additional context for each tweet.

| SNSpostID | postID              | SNSUserID | content                                                                          | favoriteCount | createdTime         |
|-----------|---------------------|-----------|----------------------------------------------------------------------------------|---------------|---------------------|
| 7         | 1121560073720008704 | 1         | To close out Autism awareness month, @ABRE...                                    | 933           | 2019-04-25 23:42:13 |
| 47        | 1120868046883528704 | 1         | Clutch. @ABREG_1 drives home 2 to take the l...                                  | 2145          | 2019-04-24 01:52:21 |
| 67        | 1120508481453166592 | 1         | Who's smoother than @ABREG_1? #TakeItBack...                                     | 877           | 2019-04-23 02:03:34 |
| 108       | 1119703315397345282 | 1         | Unofficial #Astros marketing spokesperson, @A...                                 | 3170          | 2019-04-20 20:44:08 |
| 158       | 1117892268432576513 | 1         | RT @ABREG_1: Thank you Jackie! <a href="https://t.co/...">https://t.co/...</a> 0 | 0             | 2019-04-15 20:47:40 |
| 325       | 1114610992367972352 | 1         | RT @ABREG_1: Wish you were here @trvisXX h...                                    | 0             | 2019-04-06 19:29:03 |
| 341       | 1114335663388680193 | 1         | First RBI of the home slate. Courtesy: @ABRE...                                  | 836           | 2019-04-06 01:15:00 |
| 427       | 1112050016540086273 | 1         | RT @InfieldChatter: .@astros 36 Alex Bregman...                                  | 0             | 2019-03-30 17:52:39 |
| 429       | 111200627222689249  | 1         | Huuuuuge day celebrating @ABREG_1 and @JS...                                     | 9664          | 2019-03-30 14:58:50 |

## 10. How many tags are related to “astros”?

```
1  •  SELECT DISTINCT TagTagID, COUNT(TagTagID) FROM post_tag WHERE PostSNSpostID IN(
2      SELECT SNSpostID FROM post WHERE SNSUserID IN(
3          SELECT SNSUserID FROM snsuser WHERE userName = "astros"))
4      ORDER BY COUNT(TagTagID)
5      LIMIT 10
```



The screenshot shows a database query results grid titled "Result Grid". The grid displays 1 row of data from a table with columns: TagTagID and COUNT(TagTagID). The COUNT column shows a value of 688, indicating the number of tags related to the "astros" user.

| TagTagID | COUNT(TagTagID) |
|----------|-----------------|
| 1        | 688             |

## 4 VIEWS

- 1) what twitter posts get favorites more than 3?

```
1 •  CREATE
2      ALGORITHM = UNDEFINED
3      DEFINER = `root`@`localhost`
4      SQL SECURITY DEFINER
5  VIEW `tfavoritepost` AS
6      SELECT
7          `zpost`.`SNSpostID` AS `SNSpostID`,
8          `zpost`.`postID` AS `postID`,
9          `zpost`.`SNSUserID` AS `SNSUserID`,
10         `zpost`.`content` AS `content`,
11         `zpost`.`favoriteCount` AS `favoriteCount`,
12         `zpost`.`createdTime` AS `createdTime`
13     FROM
14         `zpost`
15     WHERE
16         (`zpost`.`favoriteCount` > 3)
```

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the SQL command: `1 • select * from tfavoritepost`. The main area displays a result grid titled "Result Grid". The grid has columns: SNSpostID, postID, SNSUserID, content, favoriteCount, and createdTime. There are 7 rows of data. A context menu is open on the right side of the grid, with "Result Grid" selected. Other options in the menu include "Form Editor" and "Down".

| SNSpostID | postID              | SNSUserID | content                                          | favoriteCount | createdTime |
|-----------|---------------------|-----------|--------------------------------------------------|---------------|-------------|
| 1         | 1121607683507810... | 1         | Final: Indians 2, #Astros 1                      | 111           | 2019-04-26  |
| 2         | 1121598945145147... | 1         | We know you'd like this in your living room.     | 1068          | 2019-04-26  |
| 3         | 1121582532955230... | 1         | Hang it, then bang it. *                         | 2829          | 2019-04-26  |
| 4         | 1121579802551435... | 1         | George doesn't look too frustrated anymore.      | 1453          | 2019-04-26  |
| 5         | 1121575278755471... | 1         | Special message from the @Indians to Uncle Mi... | 623           | 2019-04-26  |
| 6         | 1121567325096800... | 1         | We couldn't ask for a more perfect night.        | 658           | 2019-04-26  |
| 7         | 1121560073720008... | 1         | To close out Autism awareness month, @ABRE...    | 933           | 2019-04-25  |

2) what ins posts get favorites more than 3?

```

1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `insfavoritepost` AS
6     SELECT
7         `wpost`.`SNSpostID` AS `SNSpostID`,
8         `wpost`.`postID` AS `postID`,
9         `wpost`.`SNSUserID` AS `SNSUserID`,
10        `wpost`.`content` AS `content`,
11        `wpost`.`favoriteCount` AS `favoriteCount`,
12        `wpost`.`createdTime` AS `createdTime`
13     FROM
14         `wpost`
15     WHERE
16         (`wpost`.`favoriteCount` > 3)

```

1 • `select * from insfavoritepost|`

100% 30:1

**Result Grid** Filter Rows: Search Export:

| SNSpostID | postID  | SNSUserID | content                                             | favoriteCount | createdTime     |
|-----------|---------|-----------|-----------------------------------------------------|---------------|-----------------|
| 2.03e18   | 2       |           | Awesome pic....                                     | 72000         | 2019/4/25 20:20 |
| 3         | 2.03e18 | 2         | KJ keeps giving up the booty!!!                     | 24924         | 2019/4/25 22:21 |
| 4         | 2.03e18 | 2         | Bulpen was solid... they deserve credit, they we... | 23631         | 2019/4/25 21:56 |
| 5         | 2.03e18 | 2         | @bodaciousbearr @manintheground lol the co...       | 40103         | 2019/4/25 21:39 |
| 6         | 2.03e18 | 2         | @daxen_w was                                        | 40487         | 2019/4/25 21:26 |
| 7         | 2.03e18 | 2         | Can??t believe y??all brought him out ???           | 39973         | 2019/4/25 21:08 |
| 8         | 2.03e18 | 2         | I want my Pedro Baez Bobblehead. ????               | 35563         | 2019/4/25 20:32 |
| 9         | 2.03e18 | 2         | Ya hahv!!! @dukie11_???????                         | 71886         | 2019/4/25 19:40 |

Result Grid Form Editor

3)which twitter post get the most favorites?

```

1 • CREATE
2      ALGORITHM = UNDEFINED
3      DEFINER = `root`@`localhost`
4      SQL SECURITY DEFINER
5      VIEW `tmostfavorites` AS
6          SELECT
7              `zpost`.`SNSpostID` AS `SNSpostID`,
8              `zpost`.`postID` AS `postID`,
9              `zpost`.`SNSUserID` AS `SNSUserID`,
10             `zpost`.`content` AS `content`,
11             `zpost`.`favoriteCount` AS `favoriteCount`,
12             `zpost`.`createdTime` AS `createdTime`
13         FROM
14             `zpost`
15         ORDER BY `zpost`.`favoriteCount`
16         LIMIT 1

```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the SQL command: `1 • select * from tmostfavorites`. The main area displays a "Result Grid" showing a single row of data from the query. The columns are labeled: SNSpostID, postID, SNSUserID, content, favoriteCount, and createdTime. The data row is: 12, 1121462699408465..., 1, RT @RealJoshReddick: Met the Monster Amon..., 0, 2019-04-25. To the right of the grid, there are two buttons: "Result Grid" (selected) and "Form Editor".

4)which ins post get the most favorites?

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the SQL code for creating a view:

```
1 • CREATE
2     ALGORITHM = UNDEFINED
3     DEFINER = `root`@`localhost`
4     SQL SECURITY DEFINER
5     VIEW `insmostfavorites` AS
6         SELECT
7             `wpost`.`SNSpostID` AS `SNSpostID`,
8             `wpost`.`postID` AS `postID`,
9             `wpost`.`SNSUserID` AS `SNSUserID`,
10            `wpost`.`content` AS `content`,
11            `wpost`.`favoriteCount` AS `favoriteCount`,
12            `wpost`.`createdTime` AS `createdTime`
13        FROM
14            `wpost`
15        ORDER BY `wpost`.`favoriteCount`
16        LIMIT 1
```

```
1 • select * from insmostfavorites
```

The screenshot shows a MySQL Workbench interface. At the top, there is a command line with the query 'select \* from insmostfavorites'. Below the command line is a results grid titled 'Result Grid'. The grid has columns labeled 'SNSpo...', 'postID', 'SNSUs...', 'cont...', 'favoriteCo...', and 'createdTime'. A single row of data is shown: 13, 2.03e18, 13, 2, and 2019/4/26 10:59. The interface includes standard MySQL Workbench navigation and search tools.

| SNSpo... | postID  | SNSUs... | cont... | favoriteCo...   | createdTime |
|----------|---------|----------|---------|-----------------|-------------|
| 13       | 2.03e18 | 13       | 2       | 2019/4/26 10:59 |             |

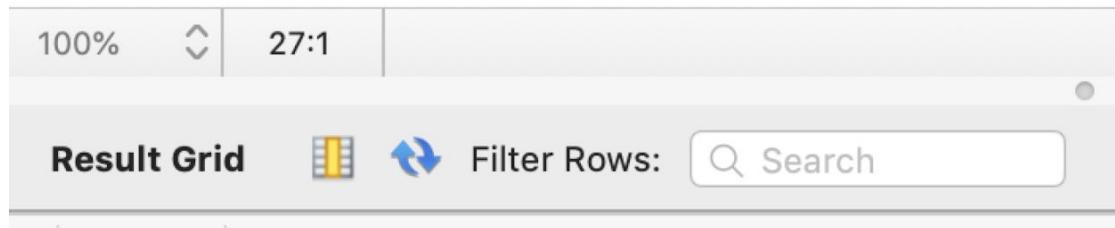
## 4 FUNCTIONS

1) get the twitter SNSuserid whose number of favorite is more than the given number.

```
CREATE FUNCTION `twitter_favorite` (favorite INT)
RETURNS INTEGER
BEGIN
SELECT SNSUserID FROM zpost where favoriteCount>=favorite
RETURN SNSUserID;
END
```

---

1 • `select twitter_favorite(8)`



2) get the ins SNSuserid whose number of favorite is more than the given number.

```
CREATE FUNCTION `ins_favorite` (favorite INT)
RETURNS INTEGER
BEGIN
SELECT SNSUserID FROM wpost where favoriteCount>=favorite
RETURN SNSUserID;
END
```

```
1 •   select ins_favorite(8)
```

3) get the twitter SNSUserID which concludes target keywords

```
CREATE FUNCTION `tcontent_target` (keywords VARCHAR(255))
RETURNS VARCHAR(255)
) BEGIN
SELECT SNSUserID FROM wpost
WHERE content like "%keywords%"
RETURN SNSUserID;
```

```
1      select tcontent_target(I)
```

100% ^ 26:1

## Result Grid

1

1

### Filter Rows:

 Search

## Export:

SNSUs...

▶ 2  
2  
2  
2  
2  
2  
2  
2

4) get the ins SNSUserID which concludes target keywords

```
CREATE FUNCTION `inscontent_target` (keywords VARCHAR(255))
RETURNS VARCHAR(255)
BEGIN
SELECT SNSUserID FROM zpost
WHERE content like "%keywords%"
RETURN SNSUserID;
END
```

1 • `select inscontent_target(I)`

## **Conclusion**

In this project, we build a database about baseball which includes the name, location and post information about baseball player, team and their fans. Our database is able to tag the social media data that we collect which can help us with the mess of social media tagging. We create some tables to define the tags which are synonyms, mis-spellings and their categories to better do the tagging work. Thus, we can know some questions like “what topics are trending in my domain” and “should I follow somebody back” after our tagging work.

## **References**

<https://tweepy.readthedocs.io/en/v3.5.0>

<https://developer.twitter.com/en/docs>

<https://instaloader.github.io/>

[https://rosettacode.org/wiki/Levenshtein\\_distance#Python](https://rosettacode.org/wiki/Levenshtein_distance#Python)