

# Dokumentation

Von Andre Steiger, Alexander Dittmer und Samuel Farley

## Inhaltsverzeichnis

### Inhalt

Inhaltsverzeichnis.....	1
Einleitung .....	2
Aufbau.....	2
Benutzer .....	3
Admin .....	3
User .....	3
Views.....	3
UserPostsView .....	3
All_user_activity.....	4
Search_Hashtag .....	4
Search_Hashtag .....	4
SentimentView.....	5
HashtagsView.....	5
HashtagsTOP20View .....	5
Triggers.....	6
trg_UpdatePost.....	6
trg_UniqueHashtagsOnPosts .....	6
Prozeduren.....	7
AddUser .....	7
DeleteUserAndRelatedData.....	7
CalculateAndInsertSentiment .....	7
Funktionen .....	8
GetUserActivityByUsername .....	8
GetAllUserActivity .....	8
GetUserActivityCount .....	9
sf_Anzahl_comments.....	9
Datenimport.....	10

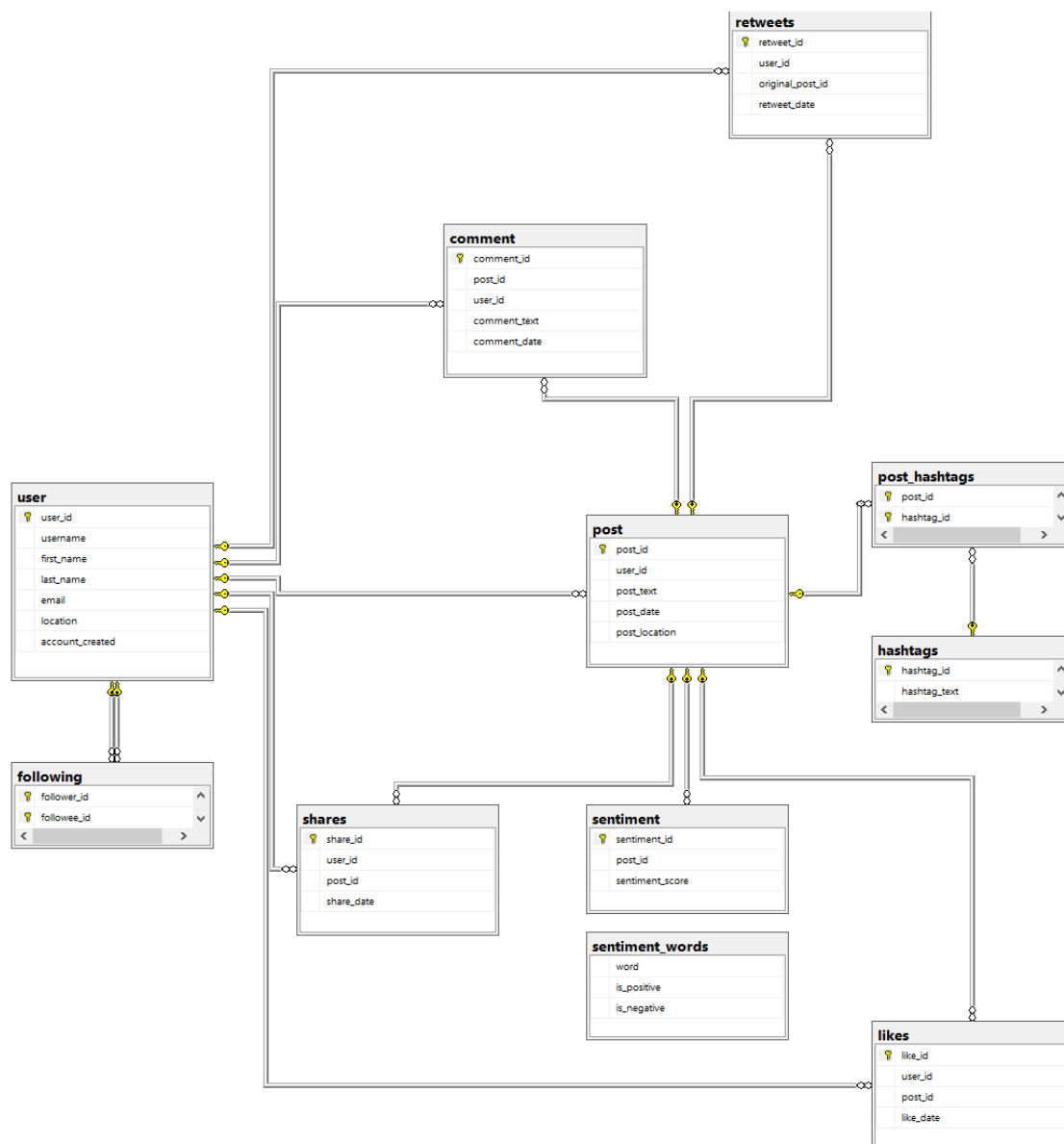
## Einleitung

Die von uns entwickelte Datenbank dient der Analyse von Meinungsäußerungen auf der Social-Media-Plattform Twitter. Sie ermöglicht insbesondere die Untersuchung von Hashtags und themenspezifischen Diskussionen. Es ist zu beachten, dass die zugrunde liegenden Daten maschinell generiert wurden.

Hinweise zur

Beschaffung und Nutzung echter Twitter-Datensätze finden sich im abschließenden Abschnitt dieses Dokuments.

## Aufbau



Im obigen Diagramm wird die Datenbank als Ganzes dargestellt, dazu auch jegliche Interaktionen und Beziehungen der Tabellen.

Bitte beachten Sie, dass die Tabelle sentiment\_words für die Prozedur [CalculateAndInsertSentiment](#) benötigt wird und daher nicht direkt mit anderen Tabellen interagiert.

## Benutzer

Für unsere Datenbank haben wir zwei Benutzer angelegt. Diese heißen Admin und User. Der User hat nur Leserechte und der Admin hat alle Berechtigungen. Die Login daten dieser Beispielaccounts lauten:

### Admin

Username: Admin

Passwort: admin

### User

Username: User

Passwort: user

## Views

### UserPostsView

Zeigt alle Posts. Gibt Username, PostID, den Text des Posts und das Datum des Posts zurück.

Beispiel:

```
SELECT *  
FROM UserPostsView  
WHERE username = 'Adria C. Manning';
```

Resultat:

	username	post_id	post_text	post_date
1	Adria C. Manning	238	die Gruppierung bezieht. good HAVING-Klausel: i...	2020-09-25 00:00:00.000

## All\_user\_activity

Zeigt die Anzahl aller Aktivitäten eines Benutzers. Dies gibt Likes, Shares, Comments, Retweets und Posts zurück.

Beispiel:

```
SELECT *  
FROM All_user_activity  
ORDER BY post_count + retweet_count + share_count + comment_count + like_count  
DESC;
```

Resultat:

	user_id	username	post_count	retweet_count	share_count	comment_count	like_count
1	80	Kenyon G. McClain	3	3	3	1	1
2	118	Tanek U. Chase	4	3	1	1	1
3	173	Imani W. Odom	2	2	1	3	2
4	265	Channing B. Wolfe	2	0	4	2	2
5	310	Sylvia W. Cotton	2	3	1	3	1
6	335	Zelenia Y. Wade	2	1	2	3	2
7	370	Cameron T. Grant	3	0	2	4	1

## Search\_Hashtag

Zeigt alle Posts unter einem Hashtag. Gibt die PostID, Hashtag, UserID, Username und den Text des Posts zurück.

Beispiel:

```
SELECT *  
FROM Search_Hashtag  
WHERE hashtag_text = '#LustrousLagoon'  
ORDER BY hashtag_text;
```

Resultat:

	post_id	hashtag_text	user_id	user_username	post_text
1	100	#LustrousLagoon	371	Grady H. Carey	zweiten Ergebnismenge enthalten sind. Mehrfach v...
2	312	#LustrousLagoon	484	Phyllis K. Maxwell	fest, ob unterschiedliche Werte als einzelne Zeilen ...
3	429	#LustrousLagoon	399	Haley J. Mitchell	vorkommt. Sonst liefert SQL eine Multimenge

## SentimentView

Zeigt alle Sentiment Scores.

Beispiel:

```
select * from SentimentView
ORDER BY Sentiment_score DESC;
```

Resultat:

	sentiment_id	post_id	sentiment_score
1	459	459	6
2	448	448	3
3	442	442	3
4	295	295	3
5	493	493	3
6	264	264	3
7	281	281	2
8	26	26	2
9	53	53	2

## HashtagsView

Zeigt alle Hashtags mit entsprechender Anzahl der Verwendungen dieses Hashtags.

Beispiel:

```
SELECT * FROM HashtagsView
ORDER BY hashtag_count DESC;
```

Resultat:

	hashtag_text	hashtag_count
1	#HamonicHaven	9
2	#HalcyonHarbor	8
3	#EnchantedEvenings	7
4	#PonderousPilgrims	6
5	#AbyssalAnthem	5
6	#MirthfulMysteries	4
7	#MiraculousMeadows	4
8	#CelestialCacophony	4
9	#GlimmersOfGrace	3
10	#LustrousLagoon	3

Diese View zeigt die Top 20 meistverwendeten Hashtags.

Beispiel:

```
SELECT * FROM HashtagsTOP20View
ORDER BY hashtag_count DESC;
```

Resultat:

	hashtag_text	hashtag_count
1	#HamonicHaven	9
2	#HalcyonHarbor	8
3	#EnchantedEvenings	7
4	#PonderousPilgrims	6
5	#AbyssalAnthem	5
6	#MirthfulMysteries	4
7	#MiraculousMeadows	4
8	#CelestialCacophony	4
9	#GlimmersOfGrace	3
10	#LustrousLagoon	3
11	#EmbersOfElation	3
12	#VelvetVistas	3
13	#WaltzingWillows	3
14	#UnchartedUtopia	2
15	#Illuminated	2

## Triggers

### trg\_UpdatePost

Dieser Trigger führt die Prozedur **CalculateAndInsertSentiment** immer dann aus, wenn ein neuer Post hinzugefügt wird.

### trg\_UniqueHashtagsOnPosts

Dieser Trigger stellt sicher, dass jedes Hashtag nur ein Mal pro Post verwendet wird.

Dieser Trigger schmeißt die Fehlermeldung 50001, wenn der Post bereits das Hashtag zugewiesen bekommen hat.

## Prozeduren

### AddUser

Diese Prozedur fügt einen Benutzer hinzu, hierzu müssen folgende Werte in folgenden Formaten angegeben werden:

```
@username NVARCHAR(50),  
@first_name NVARCHAR(40),  
@last_name NVARCHAR(40),  
@email NVARCHAR(80),  
@location NVARCHAR(40),  
@account_created DATE
```

Wenn der Benutzer erfolgreich angelegt wurde, wird „User created successfully.“ Zurückgegeben.

Gibt folgende möglichen Fehler zurück:

„Username already exists. Choose a different username.“

Wird zurückgegeben sollte der Benutzer bereits existieren.

„Email address already exists. Choose a different email address.“

Wird zurückgegeben sofern die angegebene E-Mail-Adresse bereits verwendet wird

### DeleteUserAndRelatedData

Diese Prozedur löscht den Benutzer und alle ihn bezogene Daten.

Hierzu wird lediglich die UserID in folgendem Format benötigt:

```
@user_id INT
```

Gibt 'User and related data deleted successfully.' zurück sofern erfolgreich.

### CalculateAndInsertSentiment

Diese Prozedur benötigt keinen input. Wenn es ausgeführt wird werden alle Sentiments neu berechnet und upgedated.

## Funktionen

### GetUserActivityByUsername

Diese Funktion gibt alle Aktivitäten, als Tabelle des angegebenen Benutzers wieder.  
Der Benutzername des Benutzers muss angegeben werden:

```
(@username VARCHAR(50))
```

Gibt eine Tabelle wieder die alle Posts, Comments, Likes, Shares und Retweets zeigt.

Beispiel:

```
SELECT * FROM GetUserActivityByUsername('Rylee T. Parks')
```

Resultat:

	ActivityType	ActivityID	ActivityContent	ActivityDate
1	Post	238	die Gruppierung bezieht. good HAVING-Klausel: i...	2020-09-25 00:00:00.000
2	Like	66	NULL	2020-05-17 00:00:00.000
3	Like	277	NULL	2019-11-20 00:00:00.000
4	Retweet	392	NULL	2021-09-16 00:00:00.000

### GetAllUserActivity

Gibt die Aktivität aller Benutzer als Tabelle zurück. Beinhaltet auch den Post Text falls benötigt.

Beispiel:

```
SELECT * FROM GetAllUserActivity()  
ORDER BY UserName
```

Resultat:

	ActivityType	ActivityID	ActivityContent	ActivityDate	UserName
1	Post	238	die Gruppierung bezieht. good HAVING-Klausel: ist wie ...	2020-09-25 00:00:00.000	Adria C. Manning
2	Like	66	NULL	2020-05-17 00:00:00.000	Adria C. Manning
3	Like	277	NULL	2019-11-20 00:00:00.000	Adria C. Manning
4	Retweet	392	NULL	2021-09-16 00:00:00.000	Adria C. Manning
5	Retweet	28	NULL	2022-01-08 00:00:00.000	Adrienne M. Hopper
6	Like	295	NULL	2020-04-16 00:00:00.000	Adrienne M. Hopper
7	Share	28	NULL	2020-10-02 00:00:00.000	Adrienne M. Hopper
8	Comment	107	Donec egestas. Aliquam nec enim. Nunc ut erat. Sed n...	2022-01-20 00:00:00.000	Adrienne M. Hopper
9	Comment	340	Integer	2023-06-25 00:00:00.000	Aiko G. Shaffer
10	Comment	442	nibh. Aliquam ornare, libero at auctor ullamcorper, nisl ar	2021-05-11 00:00:00.000	Aiko G. Shaffer



## GetUserActivityCount

Gibt eine Tabelle mit der Gesamtanzahl an Aktivitäten aus. Gibt bei nichtexistierendem Benutzer eine NULL-Tabelle aus.

Beispiel:

```
SELECT
  u.user_id,
  u.username,
  dbo.GetUserActivityCount(u.user_id) AS activity_count_total
FROM
  [user] u
ORDER BY activity_count_total DESC;
```

Resultat:

	user_id	username	activity_count_total
1	80	Kenyon G. McClain	11
2	173	Imani W. Odom	10
3	370	Cameron T. Grant	10
4	265	Channing B. Wolfe	10
5	437	Darius V. Castro	10
6	310	Sylvia W. Cotton	10
7	118	Tanek U. Chase	10
8	335	Zelenia Y. Wade	10

## sf\_Anzahl\_comments

Diese Funktion gibt als Output die Anzahl der Kommentare eines Benutzers innerhalb eines Jahres an.

Als Input wird die UserID und das Datum genommen.

Beispiel:

```
SELECT dbo.sf_Anzahl_comments (10, '2023-07-01') AS Anzahl_Comment;
```

Resultat:

	Anzahl_Comment
1	1

## Datenimport

Generell können die Daten, die für ein solches Programm benötigt werden auf verschiedene Weisen aus dem Internet geholt werden. Hierzu kann man einerseits das nun kostenpflichtige Twitter Development-tool verwenden (kosten ca. 5000€ pro Monat), alternativ kann man die Daten direkt von der Twitter (nun X) Seite Scrapen.

Unsere Implementation finden Sie als Scraper.py.

Ein Benutzer auf Twitter hat hierzu ein folgendes Beispiel gegeben:

```
test2.py > ...
1 import pandas as pd
2 from ntscraper import Nitter
3 import mysql.connector
4 from datetime import datetime
5 scraper = Nitter()
6 > """ ...
23 """get information about user"""
24 > def get_profile_info(name, max_retries=5, instance=None) -> dict:--
39
40
41 > class DBManager:--
77
78
79
80 user_data = get_profile_info("nypost")
81 try:
82     no_of_likes = user_data['No_of_likes']
83 except KeyError:
84     no_of_likes = None
85
86 db_manager = DBManager()
87 insert_result = db_manager.insert_user_data(
88     user_data['username'],
89     user_data['name'],
90     user_data['location'],
91     user_data['joined'],
92     user_data['No_of_tweets'],
93     user_data['following'],
94     user_data['followers'],
95     no_of_likes
96 )
97

test2.py > Q get_tweets
1 import pandas as pd
2 from ntscraper import Nitter
3
4 scraper = Nitter()
5 """get tweets with information from spec. user"""
6 def get_tweets(name, modes, no) -> list:
7     #init scraper with given param.
8     tweets: str = scraper.get_tweets(name, mode = modes, number = no)
9     tweets_data: list = []
10    #start for loop to iterate through tweets
11    for tweet in tweets['tweets']:
12        data = [tweet['text'], tweet['date'], tweet['stats']['likes'], tweet['stats']['comments']]
13        tweets_data.append(data)
14    #convert list with pandas to DF
15    data = pd.DataFrame(tweets_data, columns=['text', 'date', 'No_of_Likes', 'No_of_tweets'])
16    #save as csv file
17    data.to_csv('tweetstest.csv')
18    print(data)
19    return data
20
21 """get information about user"""
22 def get_profile_info(name, max_retries=5, instance=None) -> list:
23     user = scraper.get_profile_info(name)
24     user_data = {
25         user['username'], user['name'], user['id'], user['location'], user['joined'],
26         user['stats']['tweets'], user['stats']['following'], user['stats']['followers'],
27         user['stats']['likes']]
28
29     data_frame = pd.DataFrame(user_data, columns=['username', 'name', 'user_id', 'location', 'joined', 'No_of_tweets', 'following', 'followers', 'No_of_Likes'])
30     data_frame.to_csv('usertest.csv', index=False)
31     print(data_frame)
32     return data_frame
33
34

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
0 Almost no one seems to be aware of the immense... Dec 30, 2023 · 1:25 PM UTC 162625 21021
1 View of last night's @starlink satellite deplo... Jan 4, 2024 · 1:19 AM UTC 948 219
2 Even though Internet Archive has a ton of nega... Jan 3, 2024 · 9:57 PM UTC 10856 570
3 Falcon 9's first stage has landed on Landing 2... Jan 4, 2024 · 7:04 AM UTC 5288 1118
4 Completed Yier 24 in Diablo's Abattoir of Zir... Jan 4, 2024 · 6:01 AM UTC 16353 2335
5 Tesla 2022 recap Made possible by the hard w... Jan 4, 2024 · 1:39 AM UTC 8857 1484
6 @elonmusk Elon Musk 44196397 8:12 PM · 2 Jun 2009 35650 515 168074125 39068
7 (venv) ubuntu@ubuntu:~/practice/snscraper$
```

Resultat:

```
mysql> select * from User_demo;
```

	id	username	name	location	joined	No_of_tweets	following	followers	No_of_Likes
1	1	@MrBeast	MrBeast	Follow me for a cookie	2014-04-21	6487	1960	25216323	NULL
2	2	@GDayCTF	Ryan M. Montgomery	The Gibson	2008-05-27	1133	496	46783	NULL
3	3	@GTAVI_Countdown	GTA 6 Countdown ?		2022-02-06	940	3	136186	NULL
4	4	@scavybabyyx	Scavybabyyx		2022-09-07	1	4	0	NULL
5	5	@crazyclipsonly	Crazy Clips	Worldwide	2022-12-17	1451	159	2683179	NULL
6	6	@GrumbleAlong	Epstein Client List When?	Happy Enchilada	2023-05-15	12416	878	5885	NULL
7	7	@BillAckman	Bill Ackman	New York, NY	2017-06-29	3783	680	1054730	NULL
8	8	@nypost	New York Post	New York, NY	2008-11-18	618650	10612	3062227	NULL

8 rows in set (0.01 sec)