Data Visualization Jobs Database Report

Minghao Ru 001088106 Tong Yang 001302930

INFO 6210 Final Project

Portfolio Link:

https://github.com/yangtong951019/InfoINFO6210_Data_Jobs_Final_Project

Data Visualization Job Database Report

Abstract

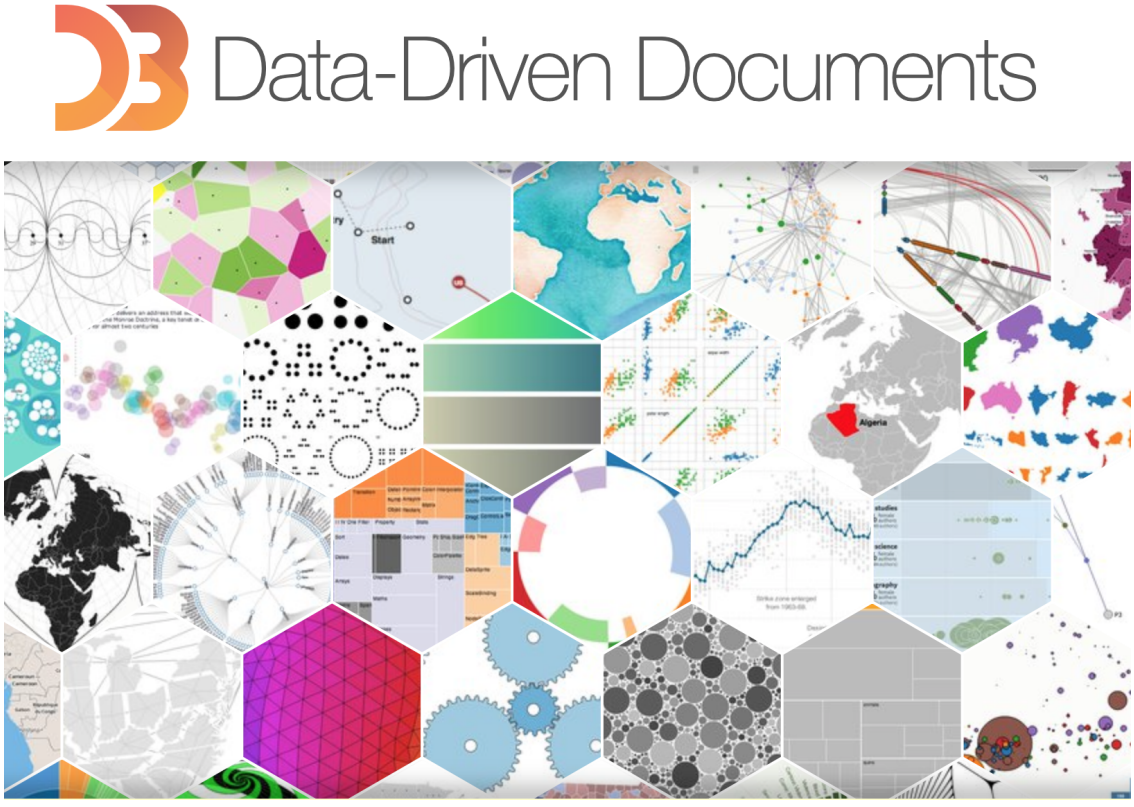
There are two parts in our projects. In the first part, we get data visualization jobs information from the website of Glassdoor based on area, i.e Boston area, California etc. After we get 10 different tables about data visualization jobs based on area, we set these tables as raw table. In part 2, we choose first three companies in Boston area, and we scrapped the information from twitter based on three companies name, employees and the jobs that the provides. Finally, we build the databases which includes ten tables, which are employee, company, twitterUser, person, tweet, tweetRhashtag, hashtag, people, peopleRjob, and Job. tweetRhashtag and peopleRjob are two relationship table, which transit information.

Key Words: Data Visualization, Jobs, Boston, Python, Sqlite3

Data Visualization Job Database Report

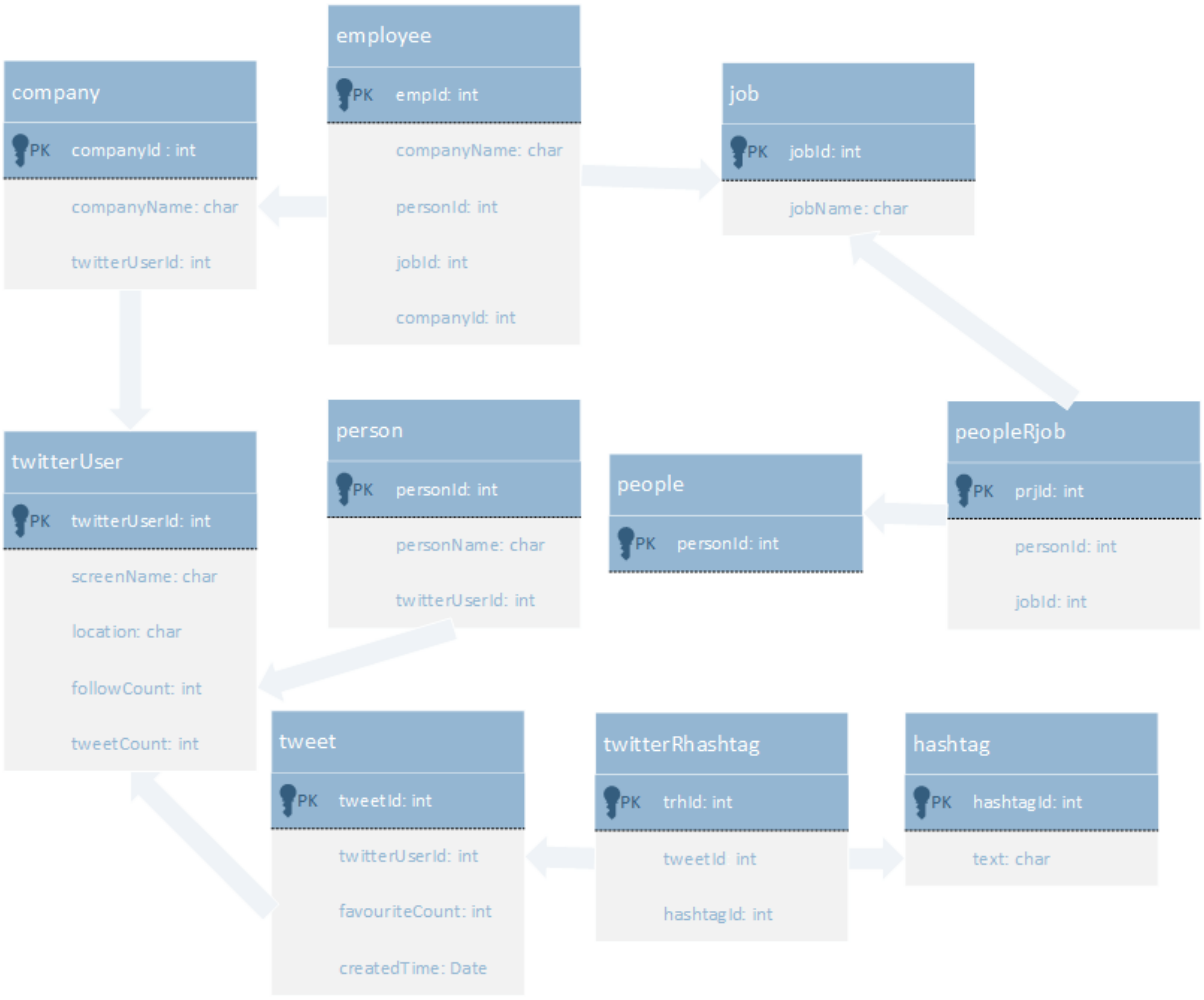
Introduction

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions. Our projects is to build a database about data visualization by scrapping information from Glassdoor and Twitter to see the trends of the job domain.



ER-Diagram

Data Visualization Job Database Report



Entities: employee, company, twitterUser, tweet, twitterRhashtag, hashtag, person, people, peopleRjob, job.

Primary Key(PK): empId, companyId, twitterUserId, tweetId, trhId, hashtagId, personId, personId, prjId, jobId

Foreign Key(FK): companyName, twitterUserId, tweetId, hashtagId, personId, jobId, companyName.

Table

company

companyId INT

companyName VARCHAR

twitterUserId INT

Data Visualization Job Database Report

people

personId INT

peopleRjob

prjId INT

personId INT

jobId INT

job

jobId INT

jobName CHAR

hashtag

hashtagId INT

text CHAR

person

personId INT

personName CHAR

twitterUserId INT

employee

empId INT

personId INT

jobId INT

companyId INT

tweet

tweetId INT

twitterUserId INT

content VARCHAR

favoriteCount INT

createdTime DATE(20)

tweetRhashtag

trhId INT

tweetId INT

hashtagId INT

twitteruser

twitterUserId INT

name CHAR

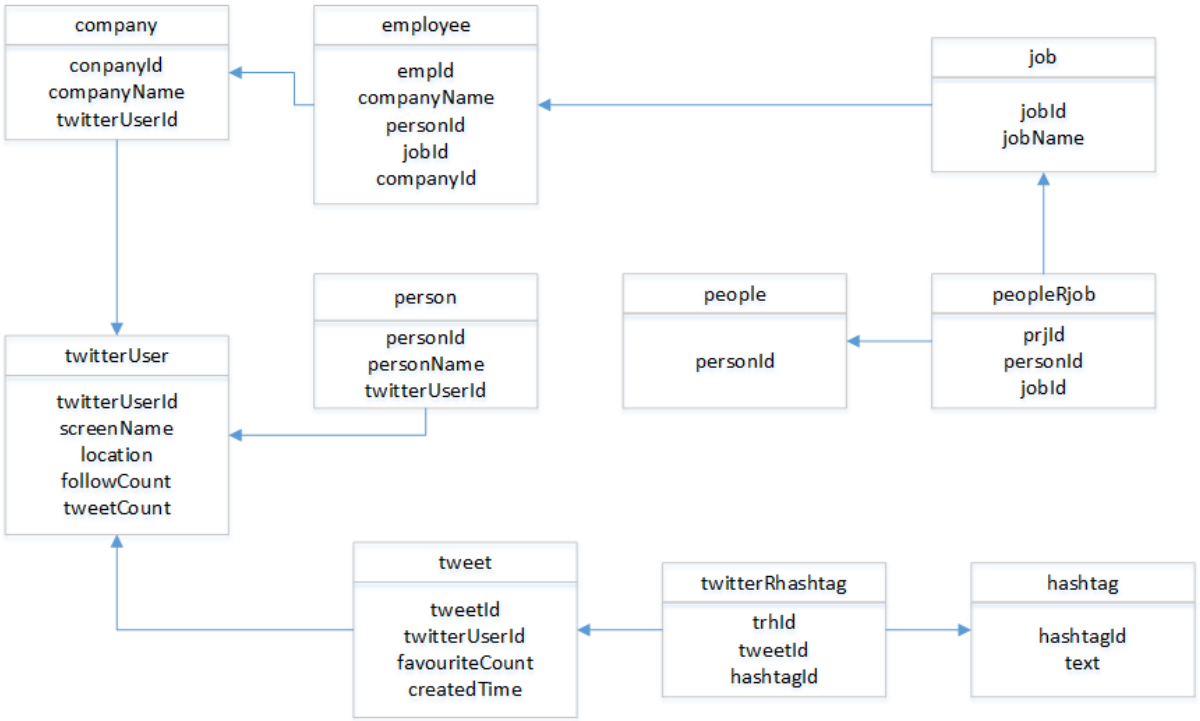
screenname CHAR

location CHAR

followersCount INT

tweetsCount INT

Conceptual Diagram



Data Collection Part

In the **first part**, we use web scrapping method to get the information of data visualization jobs based on different cities.

```
headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36 (KHTML, li
}
response = requests.get('https://www.glassdoor.com/Job/jobs.htm?suggestCount=0&suggestChosen=fals
response.status_code
```

```
200
```

```
                jobs = {
                    "Name": job_name,
                    "Company": company,
                    "State": state,
                    "City": city,
                    "Salary": salary,
                    "Location": job_location,
                    "Url": job_url
                }
                job_listings.append(jobs)
```

There are 7 columns in the table, which are jobs' name, the companies' name, the state, the city, the range of the salary, the exact location and the url of the job.

```
keyword = 'Data Visualization'
place = 'Boston'
print("Fetching job details")
scraped_data = job_listings
print("Writing data to output file")

with open('%s-%s-job-results.csv' % (place, keyword), 'wb')as csvfile:
    fieldnames = ['Name', 'Company', 'State',
                  'City', 'Salary', 'Location', 'Url']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames,quoting=csv.QUOTE_ALL)
    writer.writeheader()
    if scraped_data:
        for data in scraped_data:
            writer.writerow(data)
    else:
        print("Your search for %s, in %s does not match any jobs"%(keyword,place))
```

Here we use key word "Data Visualization" and we set the place at "Boston" to do the search, then we write data to the output file.

In the **Second Part**, we use twitter API to get information.

```
import tweepy
import json
import datetime
import warnings
warnings.filterwarnings('ignore')
consumerKey = '9oKAuFyGssLe8L9aCSTI0MRNU'
consumerSecret = 'UQJZvzAHF0no4K92nu4uoQWvt9c8PfCedNnBxMzbAlZnOe7FkF'
ACCESS_TOKEN = '520412304-CAlp1M4JafYn4LohYxqWOexB9TpmEKYqdkbz63D5'
ACCESS_SECRET = 'ijiCxaPWt0O314ljXWMywj1oiu6hvB2i42VvjQFaSfMPZ'
auth = tweepy.OAuthHandler(consumer_key=consumerKey,
    consumer_secret=consumerSecret)

#Connect to the Twitter API using the authentication
api = tweepy.API(auth)
```

We define the function appendtwitteruser(user) to search for twitter user's information of their userId, name, screenname, location, followersCount and tweetsCount. Then we use the twitter user's name of CEOs of three companies as the keywords to do the search.

```
# new twitter user appending function
# anyname to user
def anynametouser(anyname):
    user = api.get_user(anyname)
    return user
# user to user_id
def usertouserid(user):
    return user.id
# user to name
def usertoname(user):
    return user.name
# user to screen_name
def usertoscreenname(user):
    return user.screen_name
# user to location
def usertolocation(user):
    return user.location
# user to followers_count
def usertofollowerscount(user):
    return user.followers_count
# user to tweets_count
def usertotweetscount(user):
    return user.statuses_count


def appendtwitteruser(user):
    twitteruser.loc[twitteruser.shape[0]+1] = {"twitterUserId":usertouserid(user),
                                               "name":usertoname(user),
                                               "screenname":usertoscreenname(user),
                                               "location":usertolocation(user),
                                               "followersCount":usertofollowerscount(user),
                                               "tweetsCount":usertotweetscount(user)}
```

```
company.loc[company.shape[0]+1] = {"companyId":1, "c
job.loc[job.shape[0]+1] = {"jobId":1, "jobName":"Dat
person.loc[person.shape[0]+1] = {"personId":1, "pers
employee.loc[employee.shape[0]+1] = {"empId":1, "per
appendtwitteruser(anynametouser("robgremley"))
appendtwitteruser(anynametouser("etqsoftware"))
appendtwitteruser(anynametouser("DataVizSociety"))
```

```
company.loc[company.shape[0]+1] = {"companyId":2, "c
job.loc[job.shape[0]+1] = {"jobId":2, "jobName":"Dat
person.loc[person.shape[0]+1] = {"personId":2, "pers
employee.loc[employee.shape[0]+1] = {"empId":2, "per
appendtwitteruser(anynametouser("VasNarasimhan"))
appendtwitteruser(anynametouser("Novartis"))
appendtwitteruser(anynametouser("NVSData"))
```

```
company.loc[company.shape[0]+1] = {"companyId":3, "c
job.loc[job.shape[0]+1] = {"jobId":3, "jobName":"Dat
person.loc[person.shape[0]+1] = {"personId":3, "pers
employee.loc[employee.shape[0]+1] = {"empId":3, "per
appendtwitteruser(anynametouser("Bose"))
appendtwitteruser(anynametouser("Bose"))
appendtwitteruser(anynametouser("BDAnalyticsnews"))
```

Create Table Part

In the **first part**, our purpose is to create raw table that can be used for part 2. We generate information of job's name, company, state, city, salary, location and url.

Name	Company	State	City	Salary	Location	Url
Data and Vi	AllSTEM Co	IL	Chicago	$78k-$95k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=101&ao=715831&s=149&guid=00000171a4c47ef691f158443a76963e&src=GD_JOB_AD&t=SRFJ&extid=4&exst=OL&ist=&ast=
Data Engine	Sagence	IL	Chicago	$80k-$150k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=101&ao=921948&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Big Data En	Centro	IL	Chicago	$96k-$98k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=102&ao=974638&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Scienti	Mars	IL	Chicago	$94k-$119k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=103&ao=411256&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Scienti	BIOMERIEU	IL	Chicago	$70k-$118k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=104&ao=473862&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	Riskonnect	IL	Chicago	$55k-$69k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=105&ao=865989&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Senior Analy	comScore	IL	Chicago	$72k-$84k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=106&ao=800898&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Scienti	Spectrum C	IL	Chicago	$79k-$106k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=107&ao=851528&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	Wintrust Fir	IL	Chicago	$59k-$66k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=108&ao=903737&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Strategy & (	Tempus Lat	IL	Chicago	$56k-$71k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=109&ao=853928&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	Braintree	IL	Chicago	$81k-$91k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=110&ao=8095&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL&
Tableau Dat	Relativity	IL	Chicago	$56k-$108k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=111&ao=973055&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Azure Data S	Catapult Sy	IL	Chicago	$49k-$59k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=112&ao=952144&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	Kelly	IL	Chicago		Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=113&ao=935513&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	GoHealth	IL	Chicago	$73k-$82k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=114&ao=968334&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Specialist - I	Revantage	IL	Chicago	$45k-$67k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=115&ao=971289&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Strategy An	Pampered (	IL	Addison	$70k-$80k	Addison, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=116&ao=834026&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Specialist - I	Revantage	IL	Chicago	$112k-$156l	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=117&ao=970910&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Art Director	Zeno Group	IL	Chicago	$53k-$71k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=118&ao=718463&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Scienti	Strike Socia	IL	Chicago	$91k-$93k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=119&ao=4341&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL&
Research Ma	Numerator	IL	Chicago	$56k-$80k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=120&ao=841196&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	Unicorn Tec	IL	Chicago	$44k-$81k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=121&ao=389273&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Scienti	Anthem	IL	Chicago	$91k-$149k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=122&ao=133148&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	Mitchell Ma	IL	Chicago		Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=123&ao=4120&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL&
Senior Cons	Strive Cons	IL	Chicago	$68k-$102k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=124&ao=834585&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	The Scion G	IL	Chicago	$50k-$68k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=125&ao=8095&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL&
Data Scienti	SPINS, LLC	IL	Chicago	$99k-$123k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=126&ao=851062&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Digital Mark	Efinancial	IL	Chicago	$101k-$109l	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=127&ao=897610&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Data Analys	VIVA USA If	IL	Chicago	$42k-$75k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=128&ao=735551&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL
Information	Motorola Si	IL	Chicago	$42k-$80k	Chicago, IL	https://www.glassdoor.com/partner/jobListing.htm?pos=129&ao=290759&s=58&guid=00000171a4c47ef6a36bb71e53c685b8&src=GD_JOB_AD&t=SR&extid=1&exst=OL&ist=&ast=OL

In the second **part,** we created table for each entities, and we makes these table match 1NF, 2NF and 3NF normalization.

```
import pandas as pd
company = pd.DataFrame(columns=["companyId", "companyName", "twitterUserId"])
people = pd.DataFrame(columns=["personId"])
peopleRjob = pd.DataFrame(columns=["prjId", "personId", "jobId"])
job = pd.DataFrame(columns=["jobId", "jobName"])
hashtag = pd.DataFrame(columns=["hashtagId", "text"])
person = pd.DataFrame(columns=["personId", "personName", "twitterUserId"])
employee = pd.DataFrame(columns=["empId", "personId", "jobId", "companyId"])
tweet = pd.DataFrame(columns=["tweetId", "twitterUserId", "content", "favoriteCount", "createdTime"])
tweetRhashtag = pd.DataFrame(columns=["trhId", "tweetId", "hashtagId"])
twitteruser = pd.DataFrame(columns=["twitterUserId", "name", "screenname", "location", "followersCount", "tweetsCount"
```

Here we setup the pandas table. After we audit all the data that we scrap fron twiiter, we made these data into list, and we create tables.

```
c.execute("""CREATE TABLE company(
                companyId INTEGER(20) PRIMARY KEY,
                companyName VARCHAR(255),
                twitterUserId INTEGER(255),
                CONSTRAINT name_uni UNIQUE (companyId, twitterUserId)

                )""")
```

```
c.execute("""CREATE TABLE people(
                personId INTEGER(20) PRIMARY KEY,
                CONSTRAINT name_uni UNIQUE (personId)
                )""")
```

```
c.execute("""CREATE TABLE job(
                jobId INTEGER(20) PRIMARY KEY,
                jobName VARCHAR(255),
                CONSTRAINT name_uni UNIQUE (jobid)
                )""")
```

```
c.execute("""CREATE TABLE peopleRjob(
                 prjId INTEGER(20) PRIMARY KEY,
                 personId INTEGER(20),
                 jobId INTEGER(20),
                 CONSTRAINT name_uni UNIQUE (prjId)
                 )""")
```

```
c.execute("""CREATE TABLE hashtag(
                    hashtagId INTEGER(20) PRIMARY KEY,
                    text VARCHAR(255),
                    CONSTRAINT name_uni UNIQUE (hashtagId, text)
                    )""")
```

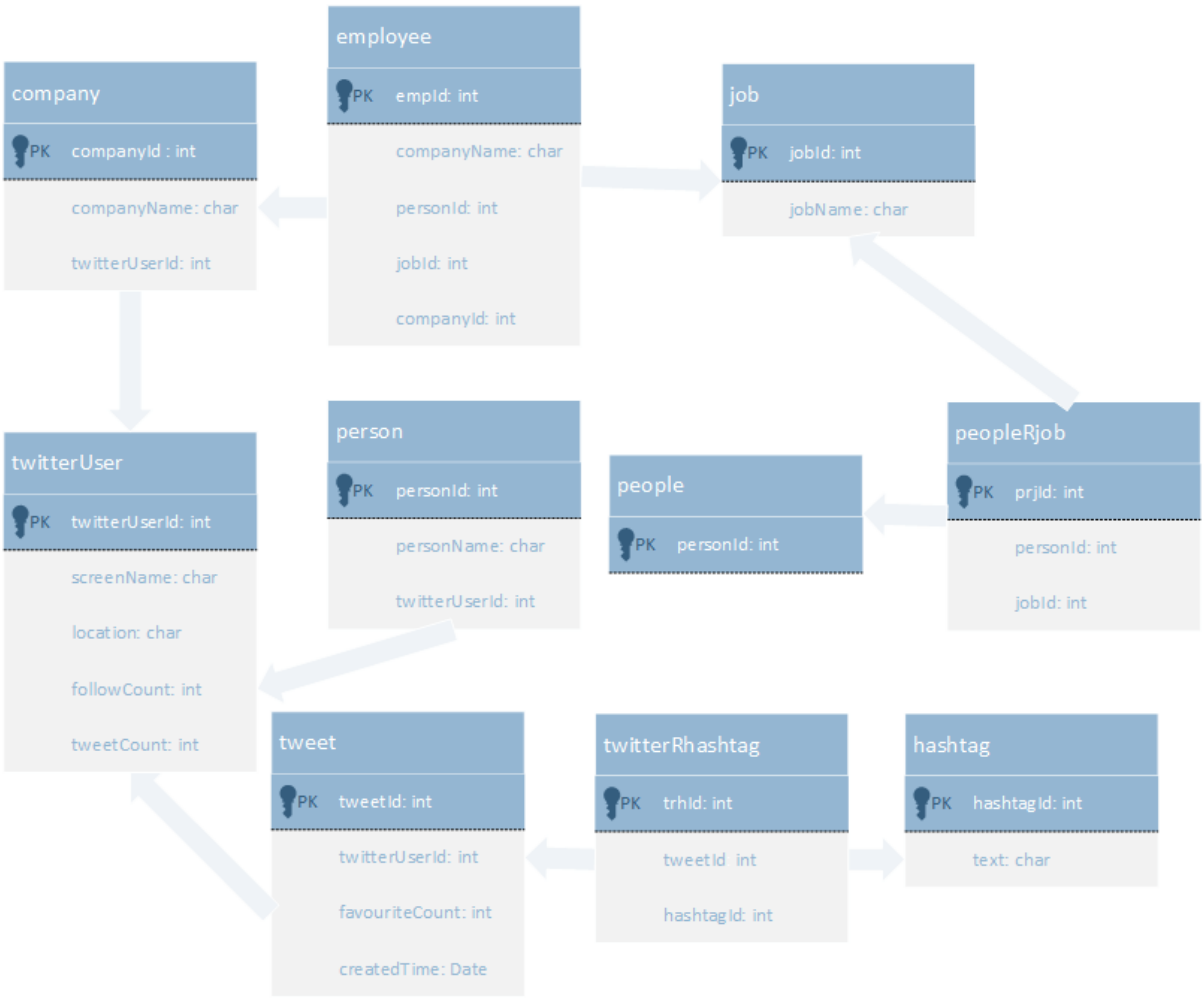
```
c.execute("""CREATE TABLE twitteruser(
                    twitterUserId INTEGER(255) PRIMARY KEY,
                    name VARCHAR(255),
                    screenname VARCHAR(255),
                    location VARCHAR(255),
                    followersCount INTEGER(20),
                    tweetsCount INTEGER(20),
                    CONSTRAINT name_uni UNIQUE (twitterUserId)
                    )""")
```

```
c.execute("""CREATE TABLE person(
                  personId INTEGER(20) PRIMARY KEY,
                  personName VARCHAR(255),
                  twitterUserId INTEGER(255),
                  CONSTRAINT name_uni UNIQUE (personId, twitterUserId)
                  )""")
```

```
c.execute("""CREATE TABLE employee(
                     empId INTEGER(20) PRIMARY KEY,
                     personId INTEGER(20),
                     jobId INTEGER(20),
                     companyId INTEGER(20),
                     CONSTRAINT name_uni UNIQUE (empId)
                     )""")
```

```
c.execute("""CREATE TABLE tweet(
                    tweetId INTEGER(255) PRIMARY KEY,
                    twitterUserId INTEGER(255),
                    content VARCHAR(255),
                    favoriteCount INTEGER(20),
                    createdTime DATE(20),
                    CONSTRAINT name_uni UNIQUE (tweetId)
                    )""")
```

```
c.execute("""CREATE TABLE tweetRhashtag(
                    trhId INTEGER(20) PRIMARY KEY,
                    tweetId INTEGER(255),
                    hashtagId INTEGER(20),
                    CONSTRAINT name_uni UNIQUE (trhId)
                    )""")
```



Here we see how each table match the normalization rules.

For 1NF, each table has a primiary key, which is a minimal set attributes which can uniquely identify a record. For company table, the primary key is company id, person ID is the primary key for people, prjID is the primary key for peopleRjob, job ID is the primary key for job table, hashtagID is the primary key for hastag table, personID is the primary key for person table, empID is the primary key for employee table, tweetID is the primary key

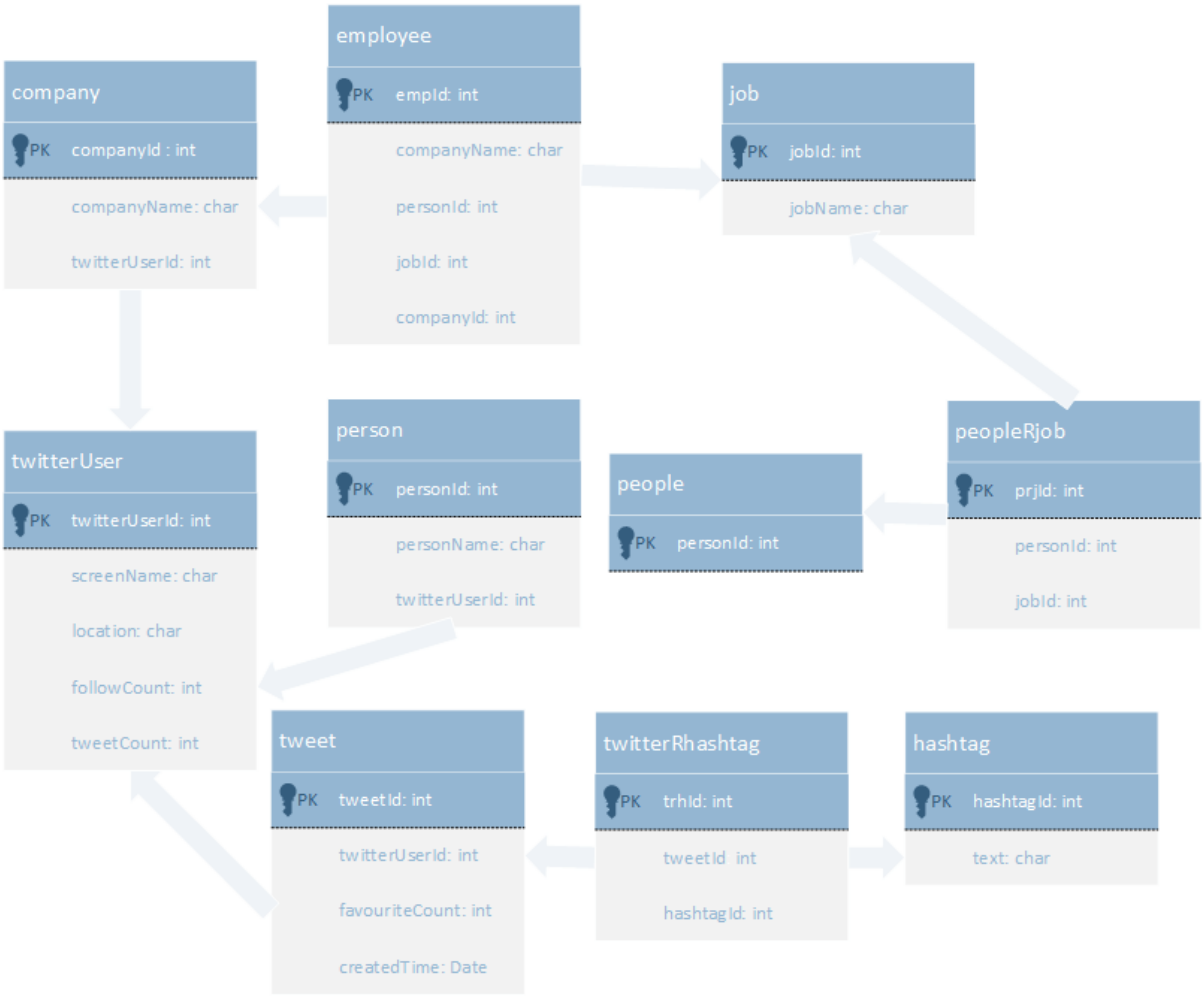
for tweet table, trhID is the primary key of tweetRhashtag table and twitterUserId is the primarykey of twitteruser table.

In each table, the value of each column are atomic, and there are no repeating groups.

For 2NF, there are no partial dependencies in each table and there are no calculated table.

For 3NF, there is no transitive denpendencies.

Database part



In the database we build, we have 10 different table, and 8 of them are entities, which the other 2 of them are transition tables. 8 entities are employee, company, twitterUser, person, tweet, hashtag, people, job. 2 transition entities are peopleRjob and twitterRhashtag. For instance, if we want to get the hashtag text from tweetId, we can use tweetID from tweet

table to lead us to twitterRhastag table, and we get the trhID based on our tweetID, then we get hashtagID based on trhID, then we use hastagID leas us to the hashtag table to find the text.

Table Audit/Clean Part

```
tweet.shape
```

```
(303, 5)
```

We see if there are any tweet with null value

```
tweet.isnull().values.any()
```

```
True
```

We see which part has null value

```
tweet.isnull().sum()
```

```
tweetId           0
twitterUserId     0
content          20
favoriteCount     0
createdTime       0
dtype: int64
```

We see the tweet with true value

```
tweet[tweet.isnull().content == True]
```

	tweetId	twitterUserId	content	favoriteCount	createdTime
284	1253707581324836864	3229087194	None	1	2020-04-24 15:29:14
285	1253700232774332419	1013777288335413248	None	0	2020-04-24 15:00:02
286	1253696980553007105	1143528514550927361	None	0	2020-04-24 14:47:07
287	1253690889941245957	1215874082	None	0	2020-04-24 14:22:55
288	1253685119308660737	286178174	None	1	2020-04-24 13:59:59
289	1253683856579969024	183263788	None	0	2020-04-24 13:54:58
290	1253681480070545409	1178423913757450253	None	0	2020-04-24 13:45:31
291	1253677664323461120	39560237	None	5	2020-04-24 13:30:22
292	1253671846920294400	85262756	None	0	2020-04-24 13:07:15
293	1253670167869100038	1200799908	None	4	2020-04-24 13:00:34
294	1253660149094498305	140743651	None	0	2020-04-24 12:20:46
295	1253659604300390400	737142202481016832	None	0	2020-04-24 12:18:36
296	1253657958711324673	715797742036062208	None	4	2020-04-24 12:12:03
297	1253644922789347328	919744697551253505	None	0	2020-04-24 11:20:15
298	1253640471307583489	19539923	None	0	2020-04-24 11:02:34
299	1253639953373753347	985425300292022273	None	0	2020-04-24 11:00:31
300	1253639846196895745	23069519	None	2	2020-04-24 11:00:05
301	1253622473284653056	14472695	None	0	2020-04-24 09:51:03
302	1253609645551255553	784790730	None	2	2020-04-24 09:00:05
303	1253569447924764672	767675409336897536	None	0	2020-04-24 06:20:21

We delete the tweet with null value in content part

```
]: tweet=tweet[tweet.isnull().content == False]
tweet.head()
```

]:

	tweetId	twitterUserId	content	favoriteCount	createdTime
1	1253396040213987329	90446356	RT @FeatCustomers: Congratulations, @etqsoftwa...	0	2020-04-23 18:51:17
2	1250886805999804429	90446356	RT @SalesTechStar: @etqsoftware State Of Quali...	0	2020-04-16 20:40:29
3	1250126049800790016	90446356	RT @AbbottNews: UPDATE: We continue to expand ...	0	2020-04-14 18:17:31
4	1250125888785612804	90446356	RT @lumileds: Great work! This simple (and fun...	0	2020-04-14 18:16:52
5	1250042422828118016	90446356	Webinar: Join ETQ Tues., 4/14, 11:00 a.m. PT f...	0	2020-04-14 12:45:12

```
]: #make tweet a csv
tweet.to_csv(r'tweet.csv',index=False, encoding="utf-8-sig")
```

Test Cases and Questions.

The first test case is to show all the companies.

Select all companies

```
: c.execute("""SELECT *
               FROM company
            """)
for row in c.fetchall():
    print(row)
```

```
(1, 'EtQ', 90446356)
(2, 'Novartis', 17226612)
(3, 'Bose', 204074545)
```

The second test case is to select all the employees

Select all employees

```
: c.execute("""SELECT *
               FROM employee
            """)
for row in c.fetchall():
    print(row)
```

```
(1, 1, 1, 1)
(2, 2, 2, 2)
(3, 3, 3, 3)
```

The third test case is to select all the screennames from twitter users.

```
c.execute("""SELECT screenname
             FROM twitteruser
          """)
for row in c.fetchall():
    print(row)
```

```
('robgremley',)
('etqsoftware',)
('DataVizSociety',)
('VasNarasimhan',)
('Novartis',)
('NVSData',)
('Bose',)
('BDAnalyticsnews',)
('TekTaurus',)
('The_AIOps_Guy',)
('femtech_',)
('manisjohal',)
('GaryGenard',)
('WebWindowsMkg',)
('ReedAbend',)
('gurobi',)
('minuskeli',)
('TietoEVRYfi',)
('abhibisht89',)
('chidambara09',)
('StartGrowthHack',)
('akdm_bot',)
('JustBeMentalist',)
('fullNam35087976',)
('StaceyMGaines',)
('mrubioc',)
('bundesbank',)
```

We generate 11 questions:

1.

Get twitteruser' userid order by their screenname in descending order

```
c.execute("""SELECT twitterUserId
             FROM twitteruser
             ORDER BY screenname DESC
          """)
for row in c.fetchall():
    print(row)
```

```
(41203376,)
(14472695,)
(85262756,)
(1215874082,)
(39560237,)
(985425300292022273,)
(1143528514550927361,)
(90446356,)
(737142202481016832,)
(784790730,)
(919744697551253505,)
(140743651,)
(183263788,)
(723488741348782080,)
(1200799908,)
(1013777288335413248,)
(3229087194,)
(715797742036062208,)
(23069519,)
(1178423913757450253,)
(17226612,)
(767675409336897536,)
(19539923,)
(286178174,)
(1086037534176473088,)
(204074545,)
(1729180992,)
```

2.

Check the when a twitteruser tweet a post and the numbers of likes of the post and the content of the post by the tweetID.

```
c.execute("""SELECT createdTime, favoriteCount, content
             FROM tweet
             WHERE tweetId = 1253396040213987329
          """)
for row in c.fetchall():
    print(row)
```

```
('2020-04-23 18:51:17', 0, "RT @FeatCustomers: Congratulations, @etqsoftware has been recognized as 'Best In Categor
y' in Spring 2020 Quality Management Software Custo…")
```

Find the numbers of posts that tweeted by user 1013777288335413248 within one day

3.

Check the when a twitteruser tweet a post and the numbers of likes of the post and the content of the post by the tweetID.

```
c.execute("""SELECT createdTime, favoriteCount, content
             FROM tweet
             WHERE tweetId = 1253396040213987329
          """)
for row in c.fetchall():
    print(row)
```

```
('2020-04-23 18:51:17', 0, "RT @FeatCustomers: Congratulations, @etqsoftware has been recognized as 'Best In Categor
y' in Spring 2020 Quality Management Software Custo…")
```

4.

Find the numbers of posts that tweeted by user 1013777288335413248 within one day

```
c.execute("""SELECT COUNT(tweetId)
             FROM tweet
             WHERE twitterUserId = 1013777288335413248    AND datetime(createdTime) > datetime('now','-1 day','localtime
          """)
for row in c.fetchall():
    print(row)
```

```
(0,)
```

5.

Find the popular posts(the favourite counts > the average numbers)

```
c.execute("""SELECT *
             FROM tweet
             WHERE favoriteCount > (SELECT AVG(favoriteCount)
                                    FROM tweet)
          """)
for row in c.fetchall():
    print(row)
```

```
tmrhyne, @yuliakrolik and @AmyCesal? Or want to catch it again? Here it is! #color #colortheory #datavisualization\nh
ttps://t.co/fj6c7ybABR', 83, '2020-04-06 21:10:33')
(1247160078651842561, 1086037534176473088, "To mark @rstudio's 20th birthday, @W_R_Chase interviewed @hadleywickham,
creator of ggplot and tidyr, about the last 20 years of R, its role in the #dataviz landscape, and what's in store fo
r the next 20 years. \n\nhttps://t.co/2dn0LSqG4X", 22, '2020-04-06 13:51:48')
(1245882381078151168, 1086037534176473088, '@tshaikh1992 @tmrhyne @alangwilson @yuliakrolik @AmyCesal Yes we'll recor
d it and post it on our YouTube channel!\nhttps://t.co/ZERxvqFOmJ', 37, '2020-04-03 01:14:41')
(1245871565134155776, 1086037534176473088, 'Color is Hard. But it doesn't have to be! Join us tomorrow at 10AM EST fo
r our first Fireside Chat. The topic is using color theory for data visualization and our panel will be @tmrhyne, @al
angwilson and @yuliakrolik hosted by @AmyCesal.\nhttps://t.co/YKPJm8XWmY https://t.co/fi2bUHkhPE', 200, '2020-04-03 0
0:31:42')
(1245014046576959493, 1086037534176473088, 'In a comprehensive and thorough deep dive, @janezhgw explores how freelan
cers can earn a living in #DataVisualization, including interviews with four designers about how they got their start
s and have built their careers: \n\nhttps://t.co/Thkvn3JW4I', 65, '2020-03-31 15:44:14')
(1244641532457746439, 1086037534176473088, 'Everything you need to know about the scatterplot, including where it cam
e from, when to use it, and how to create one. #DataVisualization \n\nhttps://t.co/ZZmw4iTw42', 40, '2020-03-30 15:0
4:00')
(1244633728007241730, 1086037534176473088, '"Flatten the Curve" has become one of the most-viewed visualizations of a
ll time. Our challenge now is to figure out how this graphic translates to real life, which means recognizing its unc
ertainty. @abmakulec:\n\nhttps://t.co/lyEEdRIfd0', 30, '2020-03-30 14:32:59')
```

6.

Find the popular posts(the favourite counts > the average numbers)

```
]:
c.execute("""SELECT *
             FROM tweet
             WHERE favoriteCount > (SELECT AVG(favoriteCount)
                                    FROM tweet)
          """)
for row in c.fetchall():
    print(row)
```

```
tmrhyne, @yuliakrolik and @AmyCesal? Or want to catch it again? Here it is! #color #colortheory #datavisualization\nh
ttps://t.co/fj6c7ybABR', 83, '2020-04-06 21:10:33')
(1247160078651842561, 1086037534176473088, "To mark @rstudio's 20th birthday, @W_R_Chase interviewed @hadleywickham,
creator of ggplot and tidyr, about the last 20 years of R, its role in the #dataviz landscape, and what's in store fo
r the next 20 years. \n\nhttps://t.co/2dn0LSqG4X", 22, '2020-04-06 13:51:48')
(1245882381078151168, 1086037534176473088, '@tshaikh1992 @tmrhyne @alangwilson @yuliakrolik @AmyCesal Yes we'll recor
d it and post it on our YouTube channel!\nhttps://t.co/ZERxvqFOmJ', 37, '2020-04-03 01:14:41')
(1245871565134155776, 1086037534176473088, 'Color is Hard. But it doesn't have to be! Join us tomorrow at 10AM EST fo
r our first Fireside Chat. The topic is using color theory for data visualization and our panel will be @tmrhyne, @al
angwilson and @yuliakrolik hosted by @AmyCesal.\nhttps://t.co/YKPJm8XWmY https://t.co/fi2bUHkhPE', 200, '2020-04-03 0
0:31:42')
(1245014046576959493, 1086037534176473088, 'In a comprehensive and thorough deep dive, @janezhgw explores how freelan
cers can earn a living in #DataVisualization, including interviews with four designers about how they got their start
s and have built their careers: \n\nhttps://t.co/Thkvn3JW4I', 65, '2020-03-31 15:44:14')
(1244641532457746439, 1086037534176473088, 'Everything you need to know about the scatterplot, including where it cam
e from, when to use it, and how to create one. #DataVisualization \n\nhttps://t.co/ZZmw4iTw42', 40, '2020-03-30 15:0
4:00')
(1244633728007241730, 1086037534176473088, '"Flatten the Curve" has become one of the most-viewed visualizations of a
ll time. Our challenge now is to figure out how this graphic translates to real life, which means recognizing its unc
ertainty. @abmakulec:\n\nhttps://t.co/lyEEdRIfd0', 30, '2020-03-30 14:32:59')
```

7.

Find the employee with most fans

```
:
.execute("""SELECT personName
            FROM person
            WHERE twitterUserId IN (SELECT twitterUserId
                                    FROM twitteruser
                                    WHERE followersCount = (SELECT MAX(followersCount)
                                                            FROM twitteruser
                                                            WHERE twitterUserId IN (SELECT twitterUserId
                                                                                    FROM person
                                                                                    WHERE personId IN (SELECT DISTINCT p
                                                                                                       FROM employee
                                                                                                       )
                                                                                    )
                                                           )
                                    AND twitterUserId IN (SELECT twitterUserId
                                                          FROM person
                                                          WHERE personId IN (SELECT DISTINCT personId
                                                                             FROM employee
                                                                            )
                                                         )
                                    )
         """)
or row in c.fetchall():
   print(row)
```

```
('Philip Hess',)
```

8.

Select the company with most fans

```
c.execute("""SELECT companyName
             FROM company
             WHERE twitterUserId IN (SELECT twitterUserId
                                     FROM twitteruser
                                     WHERE followersCount = (SELECT MAX(followersCount)
                                                             FROM twitteruser
                                                             WHERE twitterUserId IN (SELECT twitterUserId
                                                                                     FROM company
                                                                                    )
                                                            )
                                     AND twitterUserId IN (SELECT twitterUserId
                                                           FROM company
                                                          )
                                     )
          """)
for row in c.fetchall():
    print(row)
```

```
('Novartis',)
```

9.

Select the company that provide job Data Scientist

```
c.execute("""SELECT companyName
             FROM company
             WHERE companyId IN (
                                 SELECT DISTINCT companyId
                                 FROM employee
                                 WHERE jobId = (
                                                SELECT jobId
                                                FROM job
                                                WHERE jobName = "Data Scientist"
                                               )
                                 )
          """)
for row in c.fetchall():
    print(row)
```

```
('Novartis',)
```

10.

Select the top 10 locations of the Data Scientists

```
c.execute("""SELECT location, COUNT(*) AS count
             FROM twitteruser
             WHERE twitteruserId IN (
                                     SELECT twitteruserId
                                     FROM person
                                     WHERE personId IN (
                                                        SELECT personId
                                                        FROM peopleRjob
                                                        WHERE jobId = (
                                                                       SELECT jobId
                                                                       FROM job
                                                                       WHERE jobName = "Data Scientist"
                                                                      )
                                                       )
                                    )
             GROUP BY location
             ORDER BY count DESC
             LIMIT 10
          """)
for row in c.fetchall():
    print(row)
```

```
('Around the world ✈', 1)
('Berlin, Germany', 1)
('Boston', 1)
('Chiyoda-ku, Tokyo', 1)
('Devon', 1)
('Frankfurt am Main', 1)
('Helsinki', 1)
('Helsinki, Finland', 1)
('Madrid, Spain', 1)
('Mississippi', 1)
```

11.

Show the job that Bose provides.

```
]:
c.execute("""
             SELECT jobName
             FROM job
             WHERE jobId IN(
                             SELECT DISTINCT jobId
                             FROM employee
                             WHERE companyId = (
                                                SELECT companyId
                                                FROM company
                                                WHERE companyName = "Bose"
                                               )
                            )
          """)
for row in c.fetchall():
    print(row)
```

```
('Data Analytics and Insights Lead',)
```

References

https://www.tableau.com/learn/articles/data-visualization

https://stackoverflow.com/questions/35415469/sqlite3-unique-constraint-failed-error

https://www.glassdoor.com/index.htm

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

https://twitter.com/Bose

https://twitter.com/Novartis

https://twitter.com/etqsoftware

https://github.com/nikbearbrown/INFO_6210/blob/master/Movie_DB_Example/Pandas_Data_Cleaning_Wrangling_and_Visualization.ipynb

https://github.com/nikbearbrown/INFO_6210/blob/master/Examples/A_Movie_Database.ipynb

License

LICENSE:

MIT License