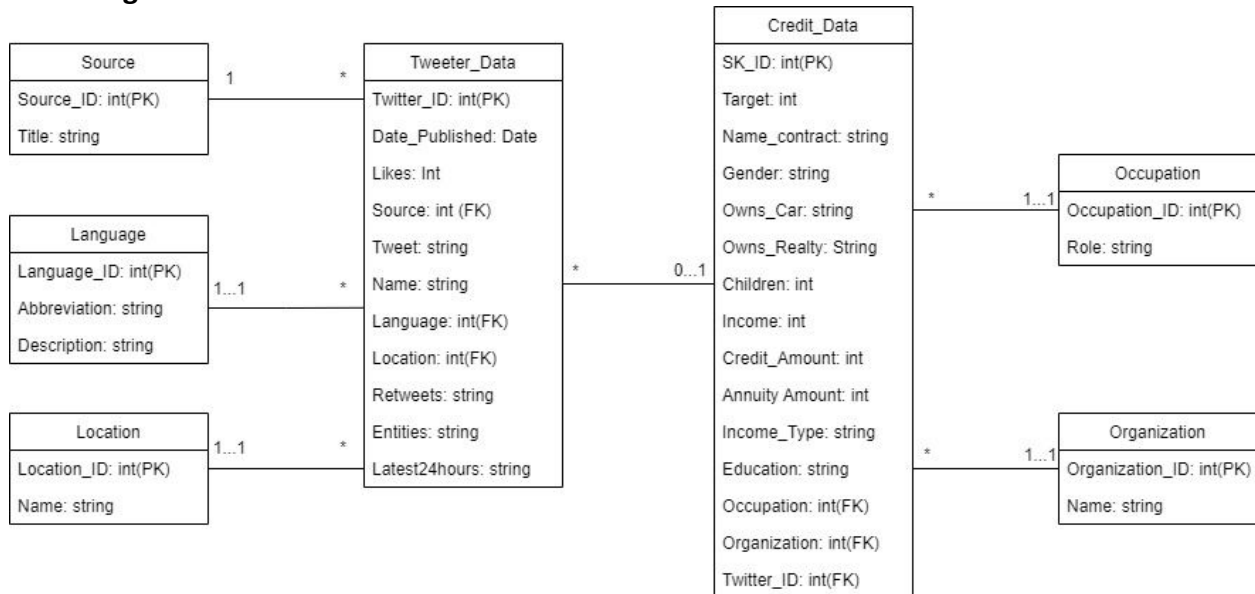# Assignment 2 – Scraping Twitter

Topic- Credit Loan Analysis
Name- Ram Bhikhalal Vaghani
NUID- 002704237

UML Diagram

1st Use Case

Description: What tweets have been liked the most?
Actor:

Precondition: Likes should be more than 1

Steps:

Actor action: Admin checks for which are the most popular category

System Responses: System will check which is the most selling point of card.

Post Condition:

Alternate Path: Finding the number of retweets

Error: No error possible

Sql –

```
#Most Liked Tweet
select twitter_id, max(likes) as Most_Liked, Tweet
from student.creditfile;
```

Relational algebra-

π twitter_id, MAX (likes) → most_liked, tweet
 γ MAX (likes) creditfile

2nd Use Case

Description: Languages in which tweets are posted
Actor:

Precondition: There should be one or more language offered in application

Steps:

Actor action: Admin checks for which are the most popular language across the application

System Responses: System can give recommendation based on the language.

Post Condition: Most popular language will display.

Error: No error possible

Sql

```
select language,count(twitter_id)
from student.creditfile
group by Language;
```

Relational algebra

γ language, COUNT (twitter_id) creditfile

3rd Use Case

Description: Most popular category in Credit card line

Precondition: Different credit cards must be provided to choose from

Actor action: Admin can update or delete the credit card according to the popularity and feedback

System Responses: System can give recommendation based on the language.

Post Condition: Most popular language will display.

Error: No error possible

Sql-

```
select tweet, max(retweets) as Retweeted
from student.creditfile;
```

Relational Algebra

π tweet, MAX (retweets) → retweeted
 γ MAX (retweets) creditfile

4th Use Case

Description: Different Application process across the platform

Precondition: There should be number of application types

Actor action: Admin can choose how the application must work

System Responses: System will show different process which is available at this point of time.

Error: May give error if important parts are left

Sql-

```
select twitter_id as Twitter_ID, count(twitter_id) as Number_of_Tweets
from student.creditfile
group by twitter_id
limit 1;
```

Relational Algebra-

π twitter_id → twitter_id, COUNT (twitter_id) → number_of_tweets
 γ twitter_id, COUNT (twitter_id) creditfile

5th Use Case

Description: Popularity of a particular credit card in different cities

Precondition: credit card should be offered in different cities

Actor action: Admin can launch new cities and can stop service too

System Responses: System will show the time left in different cities

Error: May not be able to show output if proper city is not selected

Sql-

```sql
select location, count(Twitter_id) as Total_Tweets
from student.creditfile
group by location;
```

Relational Algebra-

π location, COUNT (twitter_id) $\rightarrow$ total_tweets
 γ location, COUNT (twitter_id) creditfile

Conceptual model

CREATE TABLE `student`.`creditfile` (
  `Twitter_ID` int(20) NOT NULL,
  `Date_Published` timestamp NULL,
  `Likes` int(100) NULL,
  `Source` varchar(50) NULL,
  `Tweet` varchar(10000) NULL,
  `Name` varchar(100) NOT NULL,
  `Language` varchar(10) NOT NULL,
  `Location` varchar(30) NULL,
  `Retweets` int(50) NULL,
  `Entities` varchar(1500) NOT NULL,
  `Latest24hours` timestamp,
  PRIMARY KEY (`Twitter_ID`)
)