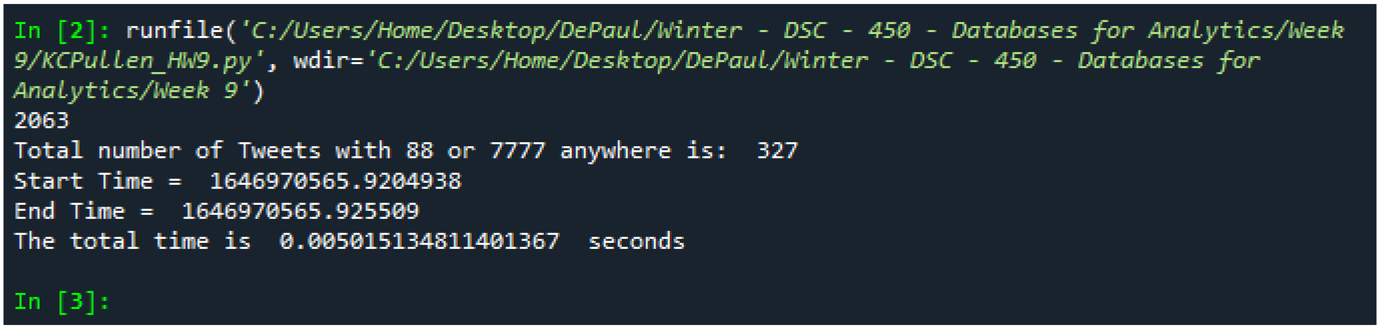
Keiland Pullen
Assignment #9

Winter 2022
DSC-450

Part 1

a.)

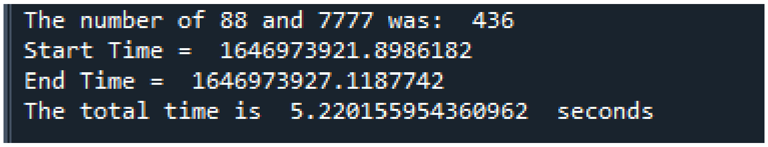
```
start = time.time()

qry = cursor.execute("SELECT * FROM Tweet where id_str LIKE '%88%' or id_str LIKE '%7777%' ")
print ("Total number of Tweets with 88 or 7777 anywhere is: ",len(qry.fetchall() ))
end = time.time()

print ("Start Time = ", start)
print ("End Time = ",end)
print ("The total time is ", ( end - start )," seconds")
```



```
In [2]: runfile('C:/Users/Home/Desktop/DePaul/Winter - DSC - 450 - Databases for Analytics/Week
9/KCPullen_HW9.py', wdir='C:/Users/Home/Desktop/DePaul/Winter - DSC - 450 - Databases for
Analytics/Week 9')
2063
Total number of Tweets with 88 or 7777 anywhere is:  327
Start Time =  1646970565.9204938
End Time =  1646970565.925509
The total time is  0.005015134811401367  seconds

In [3]:
```

b.)

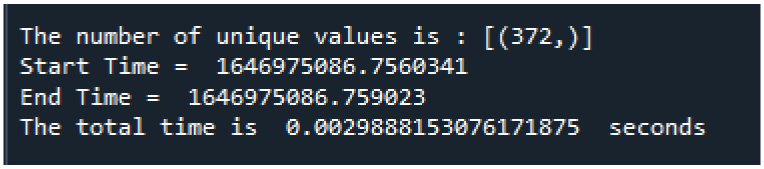
```
start = time.time()
num = 1
for i in range(lineCount):
    tweetLines = webFD.readline()
    try:
        tDict = json.loads(tweetLines.decode('utf8'))

               if ('88' in tDict['id_str']) or ('7777' in tDict['id_str']):
            # print(num)
            num = num + 1

    except ValueError:
        print("Error here")

print('The number of 88 and 7777 was: ', num)

end = time.time()

print ("Start Time = ", start)
print ("End Time = ",end)
print ("The total time is ", ( end - start )," seconds")
```



```
The number of 88 and 7777 was:  436
Start Time =  1646973921.8986182
End Time =  1646973927.1187742
The total time is  5.220155954360962  seconds
```

c.)

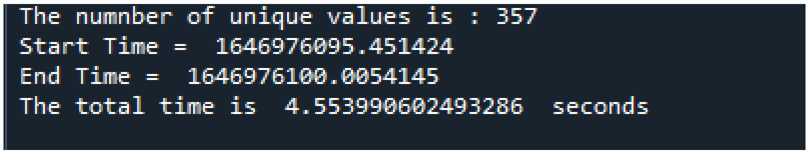
```
start = time.time()
qry2 = cursor.execute("SELECT COUNT(DISTINCT in_reply_to_user_id) From Tweet;")
qry2fetchall = cursor.fetchall()
print('The number of unique values is :',qry2fetchall)

end = time.time()
print ("Start Time = ", start)
print ("End Time = ",end)
print ("The total time is ", ( end - start )," seconds")
```



```
The number of unique values is : [(372,)]
Start Time =  1646975086.7560341
End Time =  1646975086.759023
The total time is  0.0029888153076171875  seconds
```

d.)

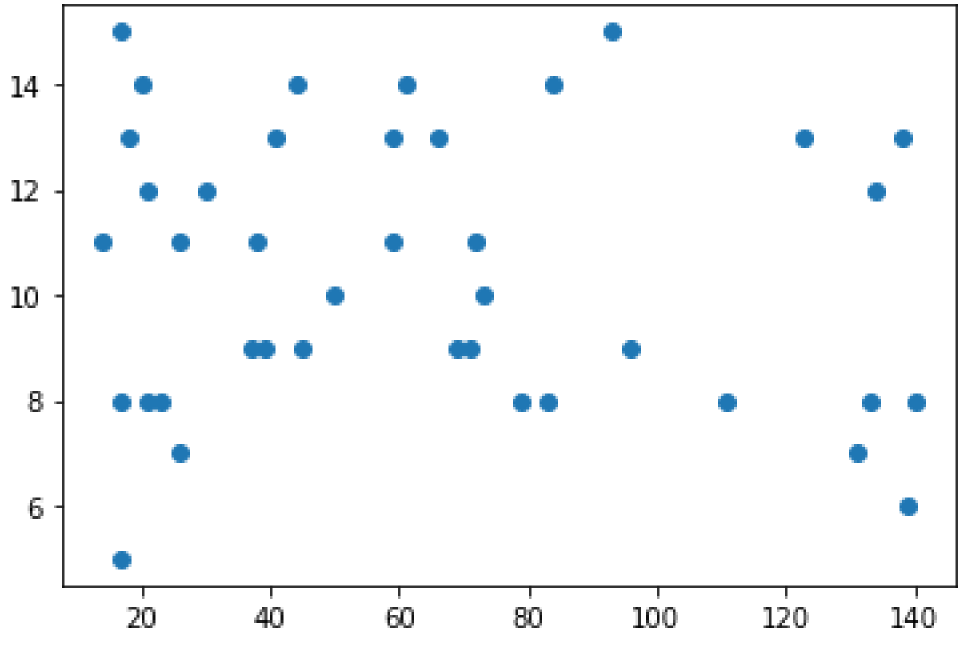
```
replyUserIdSet = set()

start = time.time()

for i in range(lineCount):
    tweetLines = webFD.readline()
    try:
            tDict = json.loads(tweetLines.decode('utf8'))

            replyUserIdSet.add(tDict['in_reply_to_user_id'])

   except ValueError:
            print("Nothing to see here")

#print(replyUserIdSet)
print ('The numnber of unique values is :', len(replyUserIdSet) )

end = time.time()
print ("Start Time = ", start)
print ("End Time = ",end)
print ("The total time is ", ( end - start )," seconds")
```



```
The numnber of unique values is : 357
Start Time =  1646976095.451424
End Time =  1646976100.0054145
The total time is  4.553990602493286  seconds
```

e.)

```
import matplotlib.pyplot as plt
Import numpy as np

txtLen = []
usrLen = []

for i in range(1, 41):
    tweetLines = webFD.readline()
    try:
        tDict = json.loads(tweetLines.decode('utf8'))

        textLength = len(tDict['text'])
        userLength = len(tDict['user']['screen_name'])

        txtLen.append(textLength)
        usrLen.append(userLength)

        # print(textLength, userLength)

    except ValueError:
        print("Nothing to see here")

x = np.array(txtLen)
y = np.array(usrLen)

plt.scatter(x, y)
plt.show()
```



Part 2.)

a.)

```
CREATE INDEX  TweetIndex
ON Tweet (userid)
```

b.)

```
CREATE INDEX UserIndex
ON User (friends_count, screen_name)
```

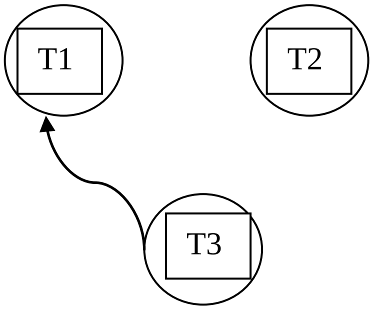
c.)

```
CREATE MATERIALIZED VIEW PartOneA AS
        SET TIMING ON;

        SELECT * FROM Tweet;
        WHERE id_str LIKE '%88%'
              OR id_str LIKE '%7777%';

        SET TIMING OFF;
```

Part 3.)

a.)

Yes, this schedule is serializable.



b.)

No, this schedule is not serializable because it contains a cycle:
T2 → T3 →T1 would be the equivalent serial schedule.