Name: Roopam Verma

Last 4 digits ID: 7590

CSE 6332, Cloud Computing

Quiz Q3, Spring 2023 (c) DL, UTA, 2023

C&P means cut and paste only those relevant lines from your program(s) into this quiz.

1. I understand that I am on my honor during this quiz, I will not collaborate, use

non-allowed sources, and I will not discuss or share this quiz with anyone for the next

12 hours.

You MAY: use Google to search, use previous source code,

YOU MAY NOT use:

Email, Facebook, Hangouts, IM, chats, Skype or ANY other human connection.

This is a timed test. Late tests will have points deducted for being late.

Very late tests will not be graded.

When you are complete, with any part, please send the link, raise your hand, so we may visually inspect that part.

The second part of the test, you should electronically submit, you will need to copy and paste

only those lines of code to implement that part of the test, usually a few (two to eight) lines of code.

Place it immediately after the question.

Submit this Quiz (renamed) with code cut and pasted, ONLY text. DO NOT submit zips, binaries, libraries,

or anything other than text.

When any parts(questions) are complete complete, submit this test, you may make multiple submissions.

If you have computer, or other, problems, please raise your hand immediately.

If you understand and agree, please initial here:

\_\_\_\_\_\_\_\_\_\_RV\_\_\_\_\_\_\_

2. Get files from this same folder.

3. Name your program/submission with your name and last digits of your ID.

4. Using the cloud services provider, all functionality possible should be implemented on that provider.

(Of course, displaying web pages through a browser and the user interface is "local")

The cloud provider is MS Azure.

5. Upload all data files (\*jpg and \*csv, not this quiz) to the cloud service.

6. Please create a relational database (your choice which) from the .csv data on the cloud provider, you may do this manually

or in code, it is your choice.

7. Please put your name and ID on each page you show for this quiz.

Show and submit code, when each part is complete, raise your hand and then show us:

For the following you may (if you wish) normalize all latitudes and longitudes to between 0 and 360 degrees, by adding 180 to all values.

10. On a web page (interface) allow a user to enter a time range (for example between 1007000 and 1007225, there were 9 events)

you will show all of the tuples (quakes) within that range, please show: lat, long, place name then time occured (time in data.)

For each show the time taken (query time) to retrieve that information, followed by the total time (depending how you do

this, these values might be the same.)

@app.route('/selectBQuery', methods=['POST'])

def selectBQuery():

time1 =request.form['time1']

time2 =request.form['time2']

# print("MAG1",type(mag1).\_\_name\_\_)

# print("MAG2",type(mag1).\_\_name\_\_)

# print(mag1,mag2)

#cursor.execute('''SELECT time,latitude, longitude, mag, place FROM [dbo].[demo\_data] where mag>='2' and mag <'5';''')

query = "select latitude, longitude, place ,time from tableName where time >= " + time1 + " or time< " + time2 +";"

print(query)

start\_time = time.time()

cursor.execute(query)

end\_time = time.time()

result = cursor.fetchall()

print(result)

time\_taken = end\_time-start\_time

#print(result)

start\_time = time.time()

#result = red.get('selectBQuery'+mag1+mag2)

if red.get('selectBQuery'+time1+time2):

result = cPickle.loads(red.get('selectBQuery'+time1+time2))

end\_time = time.time()

time\_taken = end\_time-start\_time

print("returned from cache....", result)

#cursor.execute('''SELECT time,latitude, longitude, mag, place FROM [dbo].[demo\_data] where mag>='2' and mag <'5';''')

else:

start\_time = time.time()

cursor.execute(query)

end\_time = time.time()

result = cursor.fetchall()

time\_taken = end\_time-start\_time

print("Inside....")

red.set('selectBQuery'+time1+time2,cPickle.dumps(result))

return render\_template('query2.html', tableData=result , time\_taken = time\_taken , query = query)

11. Similar to the previous question, we will give you a time range and a value Count, select exactly Count tuples from that

time range, select those at random, within that range. Note that for this example, for Count = 3, if we ask you to do

this twice, the second time should show different events. Please show quake details and time, as in the previous question.

Also note that the value Count, may be larger than the range (so some quakes will be selected more than once.)

@app.route('/searchByLatAgeRandom', methods=['POST'])

def selectCQuery():

print(request.form.get('count'))

cursor = conn.cursor()

#cursor.execute('select count(\*) from testdb.table\_1 where Latitude between '+request.form.get('lat\_1')+' and '+request.form.get('lat\_2')+' and Age between '+request.form.get('age\_1')+' and '+request.form.get('age\_2')+' ; ')

#cursor.execute('SELECT GivenName, City, State FROM testdb.table\_1 where city like \'%' + request.form.get('city') + '\' ;')

#result1 = cursor.fetchall()

beforeTime = time.time()

for x in range(1, int(request.form.get('count'))):

rand\_number = random.randrange(0, int(request.form.get('count')))

sql = 'select latitude, longitude, place ,time from tableName where time between '+request.form.get('lat\_1')+' and '+request.form.get('lat\_2')+' LIMIT {}; '.format(rand\_number)

cursor.execute(sql)

#cursor.execute('SELECT GivenName, City, State FROM testdb.table\_1 where city like \'%' + request.form.get('city') + '\' ;')

result = cursor.fetchall()

print( str(x) +' : '+ str(len(result)))

afterTime = time.time()

timeDifference = afterTime - beforeTime

return render\_template('query4.html', time=timeDifference)

12. Some of the seismometers still need calibration, they report the wrong values. The ones that under report values have a range of

time values: Min to Max (for example 1007000 to 1007225) and a known net value (for example "nc"). For any quakes within the range

specified (Min, Max, net) by a user please decrement the time value by the user supplied value dec (for example 212).

Then list the quake information for all that were modified.

Also list the count of the number of quakes modified.

Please show the time to do these modifications (similar to previous.)

13. Enable an in-memory caching mechanism (NOT an in-memory database) then:

(a) Repeat question 10.

(b) Repeat Question 11.

@app.route('/selectBQuery', methods=['POST'])

def selectBQuery():

time1 =request.form['time1']

time2 =request.form['time2']

# print("MAG1",type(mag1).\_\_name\_\_)

# print("MAG2",type(mag1).\_\_name\_\_)

# print(mag1,mag2)

#cursor.execute('''SELECT time,latitude, longitude, mag, place FROM [dbo].[demo\_data] where mag>='2' and mag <'5';''')

query = "select latitude, longitude, place ,time from tableName where time >= " + time1 + " or time< " + time2 +";"

print(query)

start\_time = time.time()

cursor.execute(query)

end\_time = time.time()

result = cursor.fetchall()

print(result)

time\_taken = end\_time-start\_time

#print(result)

start\_time = time.time()

#result = red.get('selectBQuery'+mag1+mag2)

if red.get('selectBQuery'+time1+time2):

result = cPickle.loads(red.get('selectBQuery'+time1+time2))

end\_time = time.time()

time\_taken = end\_time-start\_time

print("returned from cache....", result)

#cursor.execute('''SELECT time,latitude, longitude, mag, place FROM [dbo].[demo\_data] where mag>='2' and mag <'5';''')

else:

start\_time = time.time()

cursor.execute(query)

end\_time = time.time()

result = cursor.fetchall()

time\_taken = end\_time-start\_time

print("Inside....")

red.set('selectBQuery'+time1+time2,cPickle.dumps(result))

return render\_template('query2.html', tableData=result , time\_taken = time\_taken , query = query)

@app.route('/searchByLatAgeRandom', methods=['POST'])

def selectCQuery():

print(request.form.get('count'))

cursor = conn.cursor()

#cursor.execute('select count(\*) from testdb.table\_1 where Latitude between '+request.form.get('lat\_1')+' and '+request.form.get('lat\_2')+' and Age between '+request.form.get('age\_1')+' and '+request.form.get('age\_2')+' ; ')

#cursor.execute('SELECT GivenName, City, State FROM testdb.table\_1 where city like \'%' + request.form.get('city') + '\' ;')

#result1 = cursor.fetchall()

beforeTime = time.time()

for x in range(1, int(request.form.get('count'))):

rand\_number = random.randrange(0, int(request.form.get('count')))

sql = 'select latitude, longitude, place ,time from tableName where time between '+request.form.get('lat\_1')+' and '+request.form.get('lat\_2')+' LIMIT {}; '.format(rand\_number)

cursor.execute(sql)

#cursor.execute('SELECT GivenName, City, State FROM testdb.table\_1 where city like \'%' + request.form.get('city') + '\' ;')

result = cursor.fetchall()

print( str(x) +' : '+ str(len(result)))

afterTime = time.time()

timeDifference = afterTime - beforeTime

return render\_template('query4.html', time=timeDifference)

14. In class, on your computer, please show us the application running and we will try it out. Only for parts 10, 11, 12, 13.

15. When complete, return (send) this quiz

If you finish early, send this immediately, otherwise send between

the end of class and no more than 1 minute after that.