Name: Sushma Kukkadapu

Last 4 digits ID: 2046

Computer IP address: 129.107.80.119

CSE 6331, Cloud Computing

Quiz Q3, Spring 2019 (c) DL, UTA, 2019

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

4 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 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:

\_\_\_\_SK\_\_\_\_

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")

5. Show a web page and interface (which resides on the cloud provider) with your name and

student ID in large font at the top of every web page dispayed (for this quiz)

The cloud service provider is Microsoft Azure.

<html>

<body>

<h1> Sushma Kukkadapu </h1>

<h2> 1001642046 </h2>

</body>

</html>

Show and submit code:

6. Import the datasets into a Relational Data Base (SQL), if you have already done this, (briefly) explain/describe

how you did this. Please note that for some parts of the following you will need to create indexes (keys)

appropriately.

Imported all three csv files following way:

1. Table - codes : has the statecode.csv data

2. Table - popul : has the population.csv data

3. Table - county : has the counties.csv data

### connecting to azure cloud SQL database

connection = pyodbc.connect("Driver={ODBC Driver 17 for SQL Server};Server=tcp:sushmak.database.windows.net,1433;Database=quakes;Uid=sushma@sushmak;Pwd={azure@123};Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;")  
cursor = connection.cursor()

--------------------------------------------------------------------------------

@app.route('/')  
def index():  
 start\_time = time()  
 cursor.execute("CREATE TABLE [dbo].[codes](\  
 [code] [nvarchar](8) NULL,\  
 [state] [nvarchar](20) NULL)")  
 connection.commit()  
  
 query = "INSERT INTO dbo.codes(code,state) VALUES (?,?)"  
   
  
 with open('statecode.csv') as csvfile:  
 next(csvfile)  
 reader = csv.reader(csvfile, delimiter=',')  
 for row in reader:  
 print(row)  
 cursor.execute(query,row)  
 connection.commit()  
 end\_time = time()  
 time\_taken = (end\_time - start\_time)  
 flash('The Average Time taken to import to cloud database from csv is : ' + "%.4f" % time\_taken + " seconds")  
 cursor.close()  
 return render\_template('index.html', t=time\_taken)

-------------------------------------------------------------------------------

@app.route('/')  
def index():  
 start\_time = time()  
 cursor.execute("CREATE TABLE [dbo].[popul](\  
 [state] [nvarchar](20) NULL,\

[2010] [nvarchar](10) NULL,\

[2011] [nvarchar](10) NULL,\

[2012] [nvarchar](10) NULL,\

[2013] [nvarchar](10) NULL,\

[2014] [nvarchar](10) NULL,\

[2015] [nvarchar](10) NULL,\

[2016] [nvarchar](10) NULL,\

[2017] [nvarchar](10) NULL,\  
 [2018] [nvarchar](10) NULL)")  
 connection.commit()  
  
 query = "INSERT INTO dbo.popul(state,2010,2011,2012,2013,2014,2015,2016,2017,2018) VALUES (?,?)"  
   
  
 with open('population.csv') as csvfile:  
 next(csvfile)  
 reader = csv.reader(csvfile, delimiter=',')  
 for row in reader:  
 print(row)  
 cursor.execute(query,row)  
 connection.commit()  
 end\_time = time()  
 time\_taken = (end\_time - start\_time)  
 flash('The Average Time taken to import to cloud database from csv is : ' + "%.4f" % time\_taken + " seconds")  
 cursor.close()  
 return render\_template('index.html', t=time\_taken)

---------------------------------------------------------------------------------

@app.route('/')  
def index():  
 start\_time = time()  
 cursor.execute("CREATE TABLE [dbo].[county](\  
 [county] [nvarchar](8) NULL,\  
 [state] [nvarchar](20) NULL)")  
 connection.commit()  
  
 query = "INSERT INTO dbo.county(county,state) VALUES (?,?)"  
   
  
 with open('counties.csv') as csvfile:  
 next(csvfile)  
 reader = csv.reader(csvfile, delimiter=',')  
 for row in reader:  
 print(row)  
 cursor.execute(query,row)  
 connection.commit()  
 end\_time = time()  
 time\_taken = (end\_time - start\_time)  
 flash('The Average Time taken to import to cloud database from csv is : ' + "%.4f" % time\_taken + " seconds")  
 cursor.close()  
 return render\_template('index.html', t=time\_taken)

7. Allow a user, through a web form, to give a two letter state name (for example "TX" for Texas), and a year, and you

show the state population for that year.

8. Allow a user, through a web form, to give a two letter state name, for all counties in that state, you show count

(total, number of counties) followed by a list of all counties in that state.

9. Allow a user, through a web form, to give a year and population range, and you show the names of the states within that

population range.

10. A user will give you the number of times to do the "query", you will do the previous steps 8. and 9. that number of times

and show the results as well as the total time taken.

11. Repeat 10. using an in-memory caching mechanism (NOT an in-memory database) repeat the previous step, then show us.

12. Show GTA parts (6 not necessary), 7, 8, 9, 10, 11

(as soon as you complete any part.)

13. 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.

Good Luck!