Project 1: NoSQL

Vighnesh Sridhar

Reflection:

My approach to this project was to write three functions using the data, which include the distance function, find business based on the city function, and find businesses based on location function. The code below is my implementation of the three functions:

```
def distance(lat2, lon2, lat1, lon1):
    R = 3959
    phi1 = math.radians(lat1)
    phi2 = math.radians(lat2)
    delta_phi = math.radians(lat2 - lat1)
    lambda_phi = math.radians(lon2 - lon1)
    a = math.sin(delta_phi/2) * math.sin(delta_phi/2) + math.cos(phi1/2) * math.cos(phi2/2) * math.sin(lambda_phi/2) * math.s
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    d = R * c
    return d

    pass



def FindBusinessBasedOnCity(cityToSearch,saveLocation1,collection):
    city = filter(lambda obj: obj['city'] == cityToSearch, collection)
    #df = pd.DataFrame(city)
    #print (df)

    with open(saveLocation1, 'w') as file:
        for line in city:
            file.write(line['name'] + "$" + line['full_address'] + "$" + line['city'] + "$" + line['state'] + "\n")

    pass
```

```
def FindBusinessBasedOnLocation(categoriesToSearch, myLocation, maxDistance, saveLocation2, collection):
    lat1 = myLocation[0]
    lon1 = myLocation[1]
    #df = pd.DataFrame(collection)
    #print (df)
    categoriesToSearchSet = set(categoriesToSearch)
    #contains_category = filter(lambda obj: set(obj['categories']).issuperset(categoriesToSearchSet), collection)
    contains_category = filter(lambda obj: (set(obj['categories']) & categoriesToSearchSet), collection)
    within_distance = filter(lambda obj: distance(obj['latitude'], obj['longitude'], lat1, lon1) <= maxDistance, contains_cat

    with open(saveLocation2, 'w') as file:
        for line in within_distance:
            file.write(line['name'] + "\n")

    pass
```

I first looked at the sample.db data, which consisted of businesses and information about these businesses, including the name of the business, full address, city, state, latitude, and longitude of the business. The first function I implemented was finding all the businesses based

on a particular city to search for. I used the filter function from UnQLite. This takes in a lambda parameter, and filters the rows based on some condition specified. In this case, it takes in a collection and filters the rows based on whether the city for a row is equivalent to the city to search for. After saving this result into a collection called “city”, I wrote this to a file. I first opened the file specifying ‘w’, which means that I want to write into a file. For every row in the city collection, I wrote the name, full address, city, and state of the business into the file. I separated each component with a dollar sign delimiter.

The next function I implemented is the distance function. The distance function follows a particular method which takes in the latitude and longitude of location 1, and the latitude and longitude of location 2, and computes the distance between locations 1 and 2. First, I converted latitude and longitude 1 into radians and saved this into phi 1 and phi 2. Then, I computed delta phi, the radians of the difference between latitude 2 and latitude 1, and lambda phi, the radians of the difference between longitude 2 and longitude 1. I computed a, which is the sine of delta phi divided by 2, squared, plus the cosine of phi1 multiplied by the cosine of phi2 multiplied by the sine of delta lambda divided by 2, squared. I do another computation for c as shown in the code, and the distance is R (a constant equal to 3959) multiplied by c.

The last function I implemented finds a business based on location. It accepts categories to search, a location, and maximum distance as parameters. The task is to find all the businesses that are located within ‘maxDistance’ of ‘myLocation’. The first step I took in this function was to filter the collection to include only those rows that have the categories to search for in their ‘categories’ column. I used the same filter function as before and specified the condition to check if the intersection of the value in the ‘categories’ column in collection and the categories to search for is non-empty. I had to convert both into sets before applying the intersection function. After this, I applied the filter function to check for all the rows that fall within the maximum distance by applying the distance function on the longitudinal and latitude column, along with the latitude and longitude of the location parameter. I wrote the names of the businesses that are within the maximum distance into a file.

Lessons Learned:

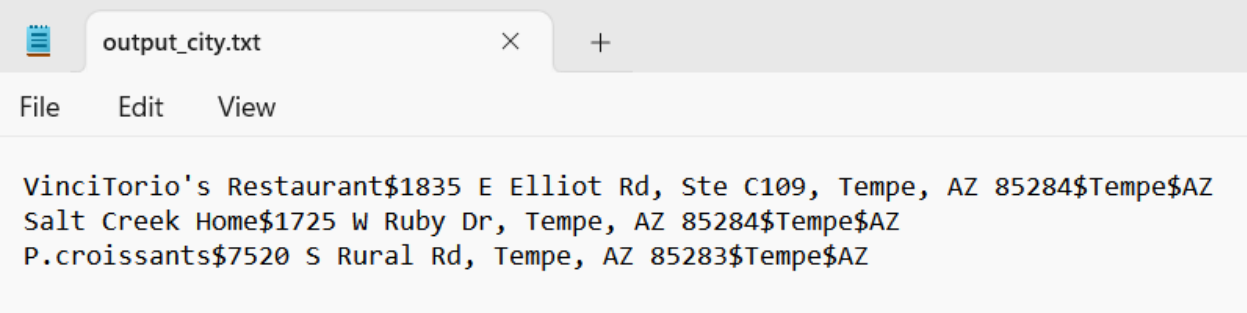
One important thing that I learned is to think about what needs to be done before starting to write code. For example, I wrote code to solve this problem using the pandas library, However, the auto grader does not have the pandas library installed, so this method would not work. Another thing I got more experienced in is how to properly debug the code. This includes putting in print statements in the appropriate locations. For example, if I was making an error, I put print statements in the appropriate locations to find out the exact line where the error was occurring.

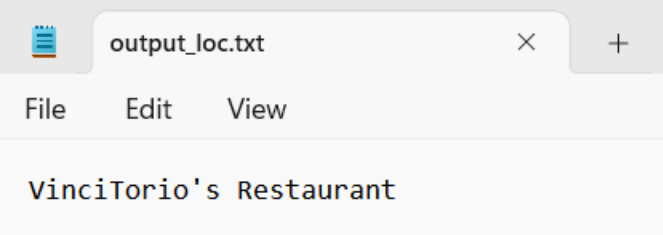
One of the errors I was getting occurred because I was using the superset or subset function on the ‘categories’ column of the collection. However, I realized that I must first convert this into a set before applying the superset function. Secondly, I realized I’m not supposed to use

superset, but rather check for an intersection. Another issue I faced was when I used the filter function. Initially I was using 'collection.filter' rather than putting collection at the end of the filter statement as shown in the code above.

Also, I learned about UnSQlite and its functionality. This includes functions such as filter and its syntax. I also gained more knowledge about writing to a file. Finally, I learned about the formula to calculate the distance between two locations given their longitude and latitude.

Output:





Result:

The first output is based on the function that finds the businesses based on the city to search for. The city to search for was Tempe. There were three businesses that are in Tempe, AZ, which are VinciTorio's Restaurant, Salt Creek Home, and P.croissants. The second output is from the function that finds businesses based on location. There was one business, VinciTorio's Restaurant, that is within 'maxDistance' of 'myLocation' and falls in the category of 'Buffets'.

```
Correct! You FindBusinessByCity function passes these test cases. This does not cover all possible test edge cases, however,
so make sure that your function covers them before submitting!


Correct! Your FindBusinessBasedOnLocation function passes these test cases. This does not cover all possible edge cases, so
make sure your function does before submitting.
```