C200 Programming Assignment Bonus

Dr. M.M. Dalkilic
Computer Science
School of Informatics, Computing, and Engineering

Indiana University, Bloomington, IN, USA

April 21, 2022

Introduction

The due date is **Thursday, April 28 at 10:59 PM EST**.

This is a bonus HW (optional) and the potential total amount of points is worth one homework (100 points). There is no unit testing accompanying this homework–and, for SQL, you'll have to read about some of the functions used that you may be unfamiliar with. You are free to either do this on your own or with a group of maximum size of four.

Queries

In class we were introduced to SQL and the relational model. You will have a great deal of freedom with this problem. Create a table called `Weather` with attributes `City`, `State`, `High`, `Low` and populate it with the data shown in Table 1. I've made equivalent list comprehension on

Weather

City	State	High	Low
Phoenix	Arizona	105	90
Tucson	Arizona	101	92
Flag Staff	Arizona	105	90
San Diego	California	77	60
Albuquerque	New Mexico	80	72
Nome	Alaska	64	-54

Table 1: Relation Weather and tuples

.

this iterable:

```
data = [('Phoenix', 'Arizona', 105, 90),
           ('Tucson', 'Arizona', 101, 92),
           ('Flag Staff', 'Arizona', 105, 90),
```

```
4            ('San Diego', 'California', 77, 60),
5            ('Albuquerque', 'New Mexico', 80, 72),
6            ('Nome', 'Alaska', 64 ,-54)]
```

You should read about these SQL functions (**NOTE**: SQL functions, not Python functions): `count()`, `sum()`, `min()`, `max()` as well as "group by" and "in". We've given the answer to query 1 (i.e. MYSQL query). The results of these queries are shown in the output. The following section shows the results obtained by using python and list comprehension-you are required to implement the MYSQL operations in python to implement these queries.

1. Select all the tuples (Query 1)

```
1 print("Query 1")
2     for i in my_cursor.execute("SELECT * FROM Weather"):
3         print(i)
4 print("List Comprehension: ", data)
```

2. Select all the tuples where the High temperature is less than 80 (Query 2)

```
1 print("Query 2")
2 print("List Comprehension: ", [d for d in data if d[2] < 80 ])
```

3. Select All the cities where the low temperature is strictly greater than the Low of Albuquerque – you cannot use the number 72.0 in the query (Query 3)

```
1 print("Query 3")
2 x =[d[0] for d in data if d[3] > [d[3] for d in data if d[0] == 'Albuquerque'][0]]
3 print("List Comprehension: ",x)
```

4. Select the city and temperature with the smallest low temperature (Query 4)

```
1 print("Query 4")
2 print("List Comprehension: ",[(d[0],d[3]) for d in data if d[3] in (sorted(data, key = lambda x:x[3])[0])])
```

5. Select the city temperature with the largest high temperature–since there are two, both cities should be returned. (Query 5)

```
1 print("Query 5")
2 print("List Comprehension: ",[(d[0],d[2]) for d in data if d[2] in (sorted(data, key = lambda x:x[2],reverse=True)[0])])
```

6. Display the *average* High and Low temperatures–you are not allowed to use Avg() (Query 6)

```
print("Query 6")
print("List Comprehension: ", [(sum([d[2] for d in data])/len(data),
    sum([d[3] for d in data])/len(data))])
```

7. Give the counts of cities by their Low temperatures (Query 7)

```
print("Query 7")
print([(i,list(map((lambda x: x[3]),data)).count(i)) for i in set(map
    ((lambda x: x[3]),data))])
```

Output

```
Query 1
('Phoenix', 'Arizona', 105.0, 90.0)
('Tucson', 'Arizona', 101.0, 92.0)
('Flag Staff', 'Arizona', 105.0, 90.0)
('San Diego', 'California', 77.0, 60.0)
('Albuquerque', 'New Mexico', 80.0, 72.0)
('Nome', 'Alaska', 64.0, -54.0)
List Comprehension:
[('Phoenix', 'Arizona', 105, 90),
 ('Tucson', 'Arizona', 101, 92),
 ('Flag Staff', 'Arizona', 105, 90),
 ('San Diego', 'California', 77, 60),
 ('Albuquerque', 'New Mexico', 80, 72),
 ('Nome', 'Alaska', 64, -54)]

Query 2
('San Diego', 'California', 77.0, 60.0)
('Nome', 'Alaska', 64.0, -54.0)
List Comprehension:
[('San Diego', 'California', 77, 60),
 ('Nome', 'Alaska', 64, -54)]

Query 3
('Phoenix',)
('Tucson',)
('Flag Staff',)
List Comprehension:  ['Phoenix', 'Tucson', 'Flag Staff']
```

Output

```
Query 4
('Nome', -54.0)
List Comprehension:  [('Nome', -54)]

Query 5
('Phoenix', 105.0)
('Flag Staff', 105.0)
List Comprehension:  [('Phoenix', 105), ('Flag Staff', 105)]

Query 6
(88.66666666666667, 58.333333333333336)
List Comprehension:  [(88.66666666666667, 58.333333333333336)]

Query 7
(-54.0, 1)
(60.0, 1)
(72.0, 1)
(90.0, 2)
(92.0, 1)
List Comprehension:  [(72, 1), (-54, 1), (60, 1), (90, 2), (92, 1)]
```

SQL

- Write the MYSQL code using python for queries 1-7.
- Note: To reiterate, you must first successfully create the table, then comment out the code for table creation or it'll throw an error. A query should be run after the table creation and populating it with data.