C200 Programming Assignment Bonus

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

Indiana University, Bloomington, IN, USA

April 21, 2023

Introduction

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

This is our last HW and carry the same points as previous HWs. For SQL, you'll have to read about some of the functions used that you may be unfamiliar with. The programming partners are given at the end of the document.

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 `Pizza` with attributes `Name`, `Size`, `Toppings`, `Beverage`, `Cost` and populate it with the data shown in Table 1. I've made equivalent list com-

Pizza

Name	Size	Toppings	Beverage	Cost
Mother Bears	large	3	2	20
Dominos	small	1	3	12
Pizza hut	large	3	2	22
Toppers	medium	2	1	15
Avers	large	2	3	24
Lennys	medium	3	4	27

Table 1: Names of Pizza places, number of toppings, beverage options and cost

.

prehension on this iterable:

```
1     data = [
2         ('Mother Bears', 'large', 3.0, 2.0, 20.0),
3         ('Dominos', 'small', 1.0, 3.0, 12.0),
4         ('Pizza hut', 'large', 3.0, 2.0, 22.0),
```

```
5          ('Toppers', 'medium', 2.0, 1.0, 15.0),
6          ('Avers', 'large', 2.0, 3.0, 24.0),
7          ('Lennys', 'medium', 3.0, 4.0, 27.0)
8      ]
```

We have taken the names of pizza places from the town of Bloomington, but we are not trying to show the exact details of how many toppings or beverage options are offered in these places. We are interested in finding the answers to our questions related to this data.

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 should implement the MYSQL operations in python to implement these queries**.

We have already solved the query-1 for you i.e, we already give the SQL code to solve query1.

1. Select all the tuples (Query 1)

```
temp = []
for i in db_cursor.execute("SELECT * FROM Pizza"):
    print(i)
print("List Comprehension: ", data)
```

2. Select all pizza options that cost less that $20.00 (Query 2)

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

3. Select all pizza places where toppings cost more than the toppings at Domino's (Query 3)

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

4. Select the pizza place and beverage with the highest cost (Query 4)

```
print("Query 4")
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 Pizza place and Cost with the smallest number of toppings (Query 5)

```
print("Query 5")
print("List Comprehension: ",[(d[0], d[4]) for d in data if d[0] in (sorted(data, key = lambda x:x[2])[0])])
```

6. Display the average number of Toppings and Beverages. You can not 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 Pizza places by their number of Toppings (Query 7)

```
print("Query 7")
print("List Comprehension: ", [(i, list(map((lambda x: x[2]),data)).count(i)) for i in set(map((lambda x: x[2]),data))])
```

Output

```
Query 1
SQL output
('Mother Bears', 'large', 3.0, 2.0, 20.0)
('Dominos', 'small', 1.0, 3.0, 12.0)
('Pizza hut', 'large', 3.0, 2.0, 22.0)
('Toppers', 'medium', 2.0, 1.0, 15.0)
('Avers', 'large', 2.0, 3.0, 24.0)
('Lennys', 'medium', 3.0, 4.0, 27.0)

List Comprehension Output:
[('Mother Bears', 'large', 3.0, 2.0, 20.0),
('Dominos', 'small', 1.0, 3.0, 12.0),
('Pizza hut', 'large', 3.0, 2.0, 22.0),
('Toppers', 'medium', 2.0, 1.0, 15.0),
('Avers', 'large', 2.0, 3.0, 24.0),
('Lennys', 'medium', 3.0, 4.0, 27.0)]

Query 2
SQL output
('Dominos', 'small', 1.0, 3.0, 12.0)
('Toppers', 'medium', 2.0, 1.0, 15.0)

List Comprehension Output:
[('Dominos', 'small', 1.0, 3.0, 12.0),
('Toppers', 'medium', 2.0, 1.0, 15.0)]

Query 3
SQL output
('Mother Bears',)
('Pizza hut',)
('Toppers',)
('Avers',)
('Lennys',)

List Comprehension Output:
['Mother Bears', 'Pizza hut', 'Toppers', 'Avers', 'Lennys']
```

Output

```
Query 4
SQL output
('Lennys', 4.0)

List Comprehension Output:
[('Lennys', 4.0)]

Query 5
('Dominos', 12.0)

List Comprehension Output:
[('Dominos', 12.0)]

Query 6
(2.3333333333333335, 2.5)

List Comprehension Output:
[(2.3333333333333335, 2.5)]

Query 7
(1.0, 1)
(2.0, 2)
(3.0, 3)

List Comprehension Output:
[(1.0, 1), (2.0, 2), (3.0, 3)]
```

SQL

- Complete the functions using MYSQL code for queries 1-7.
- Note: To reiterate, you must first successfully create the table. A query should be run after the table creation and populating it with data.

1 Programming Partners

cheng47@iu.edu, sjxavier@iu.edu
tifhuang@iu.edu, schwani@iu.edu
nokebark@iu.edu, sabola@iu.edu
jegillar@iu.edu, nkmeade@iu.edu
hollidaa@iu.edu, sjvaleo@iu.edu
bgabbert@iu.edu, wuyul@iu.edu
egillig@iu.edu, grschenc@iu.edu
sgcolett@iu.edu, jcpilche@iu.edu

jl335@iu.edu, esmmcder@iu.edu
dilkang@iu.edu, mw154@iu.edu
kbchiu@iu.edu, johwarre@iu.edu
dombish@iu.edu, criekki@iu.edu
jolawre@iu.edu, parisbel@iu.edu
jacklian@iu.edu, cjromine@iu.edu
stfashir@iu.edu, elyperry@iu.edu
jl263@iu.edu, almoelle@iu.edu
wdoub@iu.edu, lmamidip@iu.edu
siqidong@iu.edu, hnichin@iu.edu
aaberma@iu.edu, bpoddut@iu.edu
marbelli@iu.edu, imalhan@iu.edu
jasoluca@iu.edu, roelreye@iu.edu
qhodgman@iu.edu, yijwei@iu.edu
sl92@iu.edu, sowmo@iu.edu
joshbrin@iu.edu, hvelidi@iu.edu
mgambett@iu.edu, snuthala@iu.edu
nbulgare@iu.edu, pvinod@iu.edu
ljianghe@iu.edu, cartwolf@iu.edu
aketcha@iu.edu, zisun@iu.edu
jonhick@iu.edu, caitreye@iu.edu
pkasarla@iu.edu, isarmoss@iu.edu
coldjone@iu.edu, jorzhang@iu.edu
aadidogr@iu.edu, psaggar@iu.edu
brakin@iu.edu, edshipp@iu.edu
mdgamble@iu.edu, lenonti@iu.edu
moalnass@iu.edu, nireilly@iu.edu
arykota@iu.edu, alexschu@iu.edu
achimeba@iu.edu, webejack@iu.edu
nbakken@iu.edu, davthorn@iu.edu
lgflynn@iu.edu, ornash@iu.edu
arjbhar@iu.edu, bensokol@iu.edu
chnbalta@iu.edu, mostrodt@iu.edu
ecastano@iu.edu, svaidhy@iu.edu
ogift@iu.edu, jarymeln@iu.edu
joecool@iu.edu, dannwint@iu.edu
vilokale@iu.edu, kalomart@iu.edu
aladkhan@iu.edu, chnico@iu.edu
joracobb@iu.edu, cmulgrew@iu.edu
hc51@iu.edu, jwescott@iu.edu
rythudso@iu.edu, tshore@iu.edu

jaybaity@iu.edu, shethsu@iu.edu
anthoang@iu.edu, aw149@iu.edu
sijatto@iu.edu, arnpate@iu.edu
klongfie@iu.edu, dy11@iu.edu
akundur@iu.edu, jvalleci@iu.edu
dud@iu.edu, rpmahesh@iu.edu
dk80@iu.edu, vthakka@iu.edu
sihamza@iu.edu, trewoo@iu.edu
anderblm@iu.edu, dpepping@iu.edu
mgorals@iu.edu, omilden@iu.edu
abolad@iu.edu, bashih@iu.edu
tyfeldm@iu.edu, mszczas@iu.edu
georliu@iu.edu, btpfeil@iu.edu
sbehman@iu.edu, rair@iu.edu
swcolson@iu.edu, alescarb@iu.edu, zhaozhe@iu.edu
katzjor@iu.edu, ajneel@iu.edu
agoldswo@iu.edu, jyamarti@iu.edu
moesan@iu.edu, cartmull@iu.edu
neybrito@iu.edu, yz145@iu.edu
sjkallub@iu.edu, kvanever@iu.edu
bradhutc@iu.edu, crfmarti@iu.edu
ekkumar@iu.edu, ss126@iu.edu
lsherbst@iu.edu, jmissey@iu.edu
actonm@iu.edu, avpeda@iu.edu
eddykim@iu.edu, danwils@iu.edu
aalesh@iu.edu, shahneh@iu.edu
nwbarret@iu.edu, nmonberg@iu.edu
megofolz@iu.edu, sousingh@iu.edu
ivycai@iu.edu, jmom@iu.edu
sakinolu@iu.edu, leplata@iu.edu
abdufall@iu.edu, patel88@iu.edu
jackssar@iu.edu, saseiber@iu.edu
aibitner@iu.edu, macsvobo@iu.edu
jacbooth@iu.edu, yasmpate@iu.edu