COMP3311 21T1 Database Systems

Week 08

Python/Psycopg2/SQLite3

[Show with no answers] [Show with all answers]

Note: Q1-2 make use of the Psycopg2 library, while Q3-5 make use of the sqlite3 library.

1. What is the difference between a *connection* and a *cursor* in Psycopg2? How do you create each?

   [show answer]

2. [Question courtesy of Clifford Sesel] The following Python script (in a executable file called `opendb`) aims to open a connection to a database whose name is specified on the command line:

```python
#!/usr/bin/python3
import sys
import psycopg2
if len(sys.argv) < 2:
    print("Usage: opendb DBname")
    exit(1)
db = sys.argv[1]
try:
    conn = psycopg2.connect(f"dbname={db}")
    print(conn)
    cur = conn.cursor()
except psycopg2.Error as err:
    print("database error: ", err)
finally:
    if conn is not None:
        conn.close()
    print("finished with the database")
```

When invoked with an existing database, it behaves as follows

```
$ ./opendb beers2
<connection object at 0x7fac401799f0; dsn: 'dbname=beers2', closed: 0>
finished with the database
```

but when invoked with a non-existent database it produces

```
$ ./opendb nonexistent
database error:  FATAL:  database "nonexistent" does not exist
```

```
Traceback (most recent call last):
  File "./opendb", line 16, in
    if conn :
NameError: name 'conn' is not defined
```

rather than

```
$ ./opendb nonexistent
database error:  FATAL:  database "nonexistent" does not exist

finished with the database
```

What is the problem? And how can we fix it?

[show answer]

3. Using the `beers` database from Prac 05, write a Python script called `cheapest` that takes one command-line argument (beer name) and outputs the name of the bar which sells that beer at the cheapest price, as well as the price to two decimal places in parentheses. Each line should be in the format: barName ($price). If there are multiple bars selling at the same lowest price, output all of them sorted by alphabetical order.

   Some examples of use:

   ```
   $ ./cheapest many args
   Usage: ./cheapest beerName

   $ ./cheapest abc
   Invalid beerName: abc

   $ ./cheapest New
   Bar(s) where New is sold the cheapest:
   Regent Hotel ($2.20)

   $ ./cheapest Old
   Bar(s) where Old is sold the cheapest:
   Coogee Bay Hotel ($2.50)
   ```

   [show answer]

4. Still using the beers database, write a Python script called `similar-bars1` that takes two command-line arguments (bar name and N) and outputs the name of the bars which sell at least N same beers as the input bar. Each line should be in the format: barName (numSameBeers). Order the output in descending order of number of same beers sold, then by ascending order of the bar names. Try to do this in the 'less efficient' approach, that is, keeping most of the logic in Python.

Some examples of use:

```
$ ./similar-bars1 abc
Usage: ./similar-bars1 barName N

$ ./similar-bars1 fakeBar 0
Invalid barName: fakeBar

$ ./similar-bars1 'Marble Bar' 1
Bar(s) which sell at least 1 same beers as Marble Bar:
Coogee Bay Hotel (3)
Royal Hotel (3)
Regent Hotel (2)
Australia Hotel (1)
Lord Nelson (1)

$ ./similar-bars1 'Royal Hotel' 3
Bar(s) which sell at least 3 same beers as Royal Hotel:
Coogee Bay Hotel (3)
Marble Bar (3)

$ ./similar-bars1 'Royal Hotel' 5
No bars sell at least 5 same beers as Royal Hotel.
```

[show answer]

5. The approach to the previous question can certainly be improved by pushing the logic down to the DB level. Re-write the Python script for `similar-bars1` as a new 'improved' script named `similar-bars2`.

   Some examples of use:

```
$ ./similar-bars2 abc
Usage: ./similar-bars2 barName N

$ ./similar-bars2 fakeBar 0
Invalid barName: fakeBar

$ ./similar-bars2 'Marble Bar' 1
Bar(s) which sell at least 1 same beers as Marble Bar:
Coogee Bay Hotel (3)
Royal Hotel (3)
Regent Hotel (2)
Australia Hotel (1)
```

```
Lord Nelson (1)

$ ./similar-bars2 'Royal Hotel' 3
Bar(s) which sell at least 3 same beers as Royal Hotel:
Coogee Bay Hotel (3)
Marble Bar (3)

$ ./similar-bars2 'Royal Hotel' 5
No bars sell at least 5 same beers as Royal Hotel.
```

[show answer]