

SPL221: Assignment 4

Python and SQL

9/01/2022

Responsible TA: Or Dinari

Due Date: 17/1/2022 23:59

1 Introduction

In this assignment you are required to implement the database of *Pizza Hat*, a shop specialized in pizza shaped hats. The assignment is composed of three parts:

1. Create and populate the database according to a configuration file.
2. Execute a list of orders, according to a second file.
3. Print a summary in a third file.

Note that you **must** use a Persistence Layer, with DTO and DAO, as seen in PS13. Failing to use it will results in a 20 points deduction. ORM is optional.

For this assignment you will use *Python 3.6.9* and *sqlite3*, which are installed on the computers in the *computer labs*.

For configuration file *config.txt*, orders file *orders.txt*, output file *output.txt* and output db file *database.db*, you will run the code (on the lab computers) with the following line:

```
> python3 main.py config.txt orders.txt output.txt database.db
```



Before writing any code, make sure you read the **entire** assignment.

2 Building the Database

The database will contain the following tables:

2.1 Structure

- hats: Hold the information on the hats currently in the inventory.
 - id INTEGER PRIMARY KEY
 - topping STRING NOT NULL
 - supplier INTEGER REFERENCES Supplier(id)
 - quantity INTEGER NOT NULL
- suppliers: Holds the suppliers data.

- id INTEGER PRIMARY KEY
- name STRING NOT NULL
- orders: Holds the information on the different orders.
 - id INTEGER PRIMARY KEY
 - location STRING NOT NULL
 - hat INTEGER REFERENCES hats(id)

2.2 Configuration file

In order to build the database, you will parse a configuration file, the file will have the following structure:

```
<#1>,<#2>
<hats>
<suppliers>
```

Where each of the numbers in the first line stands for the number of entries of that type, and each entry has the relevant table details, separated by a comma. For example:

```
3,2
1,olives,1,10
2,mushrooms,1,20
3,mushrooms,2,10
1,Scrabbles
2,Hatters
```

In the above example we have 3 type of hats in the inventory and 2 suppliers. Note that there are no spaces, just comas and new lines (Use the attached input file, and do not copy-paste from here).

3 Orders

The orders file will have the following structure:

```
<location1>,<topping1>
<location2>,<topping2>
<location3>,<topping3>
<location4>,<topping4>
```

3.1 Executing the Orders

Each order will be executed, if a certain hat topping has two suppliers or more, the inventory from the first supplier (ordered by id) will be used. When executing the orders, you will need to update the quantity of the hats in the database, if the quantity drops to zero, the entry should be removed from the database. Executed orders will be inserted to the orders database, with a unique id, starting from 1, and increasing by 1 on each order.

4 Summary File

After each order a line will be added to the summary, the line should include:

```
<topping>,<supplier>,<location>
```

For example, using the above configuration file and the following orders:

```
Hedera,olives
Hedera,mushrooms
Tel-Aviv,mushrooms
```

The result summary file will look like:

```
olives,Scrabbles,Hedera
mushrooms,Scrabbles,Hedera
mushrooms,Scrabbles,Tel-Aviv
```

5 Submission And Testing

You may assume all input will be valid, and that there will be no illegal orders (such as ordering a topping which does not exist in the inventory).

5.1 Testing

The assignment will be tested automatically **on the LABS**, we will check **both** the output file, and the database file after finishing the execution. You **MUST** use exactly the specification above, for both the output file and the database. Failing to do so will result in the automatic tests failing, we will not allow appeals on such case.

You are supplied with an example input, expected output, and the testing files. In order to test your assignment, run the following command:

```
> python3 test_assignment.py ID1_ID2.zip config.txt orders.txt true_output.txt true_database.db
```

DO NOT USE PACKAGES WHICH ARE NOT AVAILABLE IN THE LABS, the testing is automatic, if your code will fail to run due to a missing package you will get a 0.

5.2 Submission

Your submission should include a *main.py* file which we will use to run the assignment (according to 1). You may add additional *.py* files, everything should be zipped to ID1_ID2.zip.



Make sure you **commit** the changes to the database at the end of the code execution! We will check both the database and the output file.