Software Development Tools & Systems Programming

Eleftheriadis Dimitrios

computeSales.py

The project's goal is the implementation of a program, which will accept as input receipt files and will evaluate the correctness and print statistics, in python.

SQLite Option Explanation

After a lot of research, I ended up utilizing the sqlite library provided by python. The reason was because sqlite was written in C, offering much faster speeds in searches, and in storing information even huge files.

Also, sqlite provides the ability to temporarily store the entire database in memory, which makes it even faster than read / write to disk.

For example, for a text file containing 13,217,667 lines, the program takes just 46 seconds to read, check the accuracy and store the information in the database. Recovering and displaying data from the Database takes only fractions of a second.

The program should run with the sudo command to list the necessary permissions for the database, however depending on the rights of each user may not be necessary.

This result came as an average of repeated code tests on a Dell laptop with an Intel (R) Core (TM) i7-6500U CPU @ 2.50GHz and 7860MB of total memory.

Program Functionality

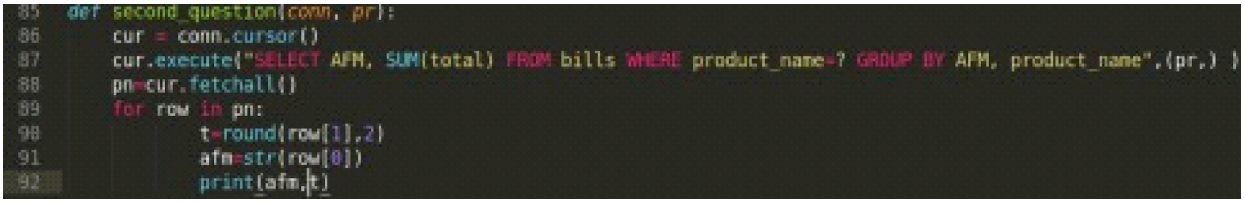
Initially, the menu was implemented with 4 options: 1 for reading a receipt file, 2 for the statistics of a product, 3 for the statistics of a VAT (ΑΦΜ) number and 4 for exit, in a while loop. Then the functionality of the 3 options was implemented. To read the receipt file:

The name input is requested in the function and subsequently the file is opened for reading. By reading line-by-line the correctness of the file is checked. When the line

that refers to the VAT number is found, it is passed through the control (it will be analyzed below). So I continue for the whole receipt and find for each product the name, the quantity, the price, the total and finally the bill total price. If the divider of the receipt was read, it means the current receipt is finished. I pass all these data to a temporary table and then if all the data are correct it is passed to the main table where all the correct receipts are stored.

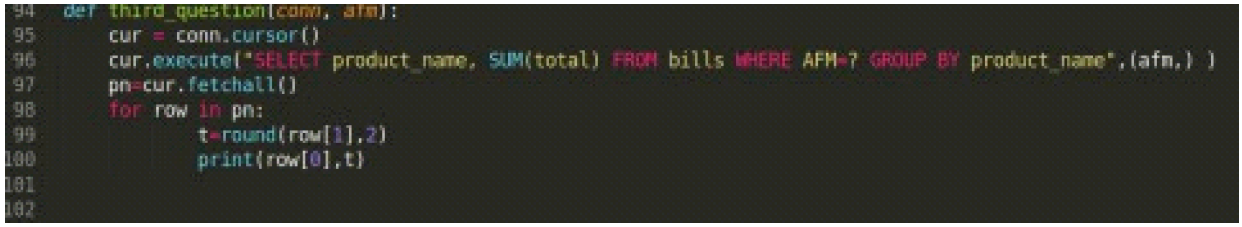
I created these tables, as the implementation of the other 2 options was done through sqlite and I created a local database. So I created a function for the connection to the local database, two functions that create the tables (one for the temp table and one for the bill table), two functions to insert the tables and a function to delete the elements from the temp table.

To print product statistics:



As mentioned above this query was implemented with sqlite. Below is the code:

To print the statistics of a VAT number:



In the same logic as the previous one, this option was implemented. The code is given:

Checks

For the implementation of the program several checks had to be made. The first test is the VAT number, i.e. if it exists and if it is 10 digits. It was also checked whether the total price of a product in proportion to the quantity is correct, but also if the final price of the receipt is also correct.
I also check if a new receipt starts, where I take the necessary measures, resetting some auxiliary variables and emptying the temp table. It is worth noting that if an error is found at any point in the receipt, then the program omits all rows of this receipt until the next one.

The program is case-insensitive using the upper () function when the user input is read for the product they want to search for.

The statistics are also printed in ascending order via group by.

Exception handling has also been utilized where necessary. An example is in the case that the price of the product is missing in a line of the receipt, this exception is caught, and the receipt is set as wrong.

Conclusions

In conclusion, python is one of the most user-friendly languages for file handling and in combination with sqlite's built-in library the query results were fast and consistent.