Luigi De Russis, Teodoro Montanaro

LAB 4 – USING DATABASES (SQLITE) WITH PYTHON

EXERCISE 1 – CREATE A SQLITE DATABASE

Perform the following actions.

1. Using SQLiteStudio, create a sqlite database (task_list.db).
2. Create the "task" table with the following columns:
   - id_task: it will contain an (auto generated) integer value that represents the unique identifier of each task;
   - todo: it will contain the text of each task;
3. Insert (by hand) in the just created database all the tasks contained in the "task_list.txt" file. The file can be downloaded at the following link: https://goo.gl/6EFKnn .

Suggestions:

1. SQLiteStudio is portable, so it does not need any installation and can be downloaded at the following link: https://sqlitestudio.pl/index.rvt?act=download

EXERCISE 2 – TELEGRAM BOT: USE THE DATABASE INSTEAD OF THE TEXT FILE

Modify the Telegram bot developed in the previous laboratory[1] to increasingly substitute the text file with the database.

The following exercises will incrementally re-implement existing features so that the script will use the database instead of the file

Exercise 2 - part 1 – /showTasks: Show all existing tasks

Show all existing tasks, sorted in alphabetic order reading from the database: disable all existing options except the "/showTasks" one and modify the right method to read from the database (instead of reading from the text file).

Consequently, the Telegram bot will accept only the following command

- /showTasks

and every time the user selects it, the program will show the tasks getting them from the database.

[1] A possible solution to the exercise can be found at https://github.com/AmI-2017/python-lab3 (*AmITaskListBot.py*).

Suggestion:

When you prepare the sql query, you should use placeholders to specify parameters. Look at the following documentation to understand what is the right placeholder for sqlite databases: https://docs.python.org/3/library/sqlite3.html

Exercise 2 - part 2 – /newTask: Add a new task

Add a new task to the "task_list.db" database.

Consequently, the Telegram bot will accept the following commands:

- /showTasks
- /newTask <task to add>

and every time the user selects the "/newTask" one, the program will add the task to the database (instead of adding it to the file).

Exercise 2 - part 3 – /removeAllTasks: Remove existing

Remove all the existing tasks that contain a provided string from the database.

Consequently, the Telegram bot will accept the following commands:

- /showTasks
- /newTask <task to add>
- /removeAllTasks <substring to use to remove all the tasks that contain it>

and every time the user selects the "/removeAllTasks" command, the program will remove the tasks from the database.