

# LAB 4 – USING DATABASES WITH PYTHON

## GETTING STARTED

The goal of this set of exercises is to develop a Telegram bot that interacts with a database.

*Recap:*

1. Fork your own copy of the Git repository associated with this lab (<https://github.com/Aml-2018/python-lab4>) to your personal GitHub space
2. Open PyCharm Professional and select Checkout from Version Control > Git in the “Welcome to PyCharm” window, to clone your (forked) repository
3. Fill the requested fields (repository URL, location on disk, ...) and press the “Clone” button
4. Once the project is open, you can create a new Python file by right clicking on the project name (Project tab, on the left) and selecting New > Python File
5. Commit and push the changes you made back to GitHub, from the VCS menu in PyCharm

## EXERCISE 1 – CREATE A DATABASE

Perform the following actions.

1. By using the Database view of Pycharm, create a database to store the tasks. You can choose to create a MySQL/MariaDB or a SQLite database.
2. Create the “task” table with the following columns:
  - id\_task: an auto generated integer value that represents the unique identifier of each task;
  - todo: the text of each task;
3. Insert “by hand” all the tasks contained in the “task\_list.txt” file. The file can be downloaded at the following link: <https://goo.gl/6EFKnn> .

## EXERCISE 2 – TELEGRAM BOT WITH DATABASE

Modify the Telegram bot developed in the previous laboratory<sup>1</sup> by replacing the text file with the database. The bot should accept the same commands of the previous version:

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

<sup>1</sup> A possible solution to the exercise can be found at <https://github.com/Aml-2018/python-lab3/tree/solution> (AmlTaskListBot.py).

### **/showTasks**

Show all existing tasks, sorted in alphabetic order, by reading them from the database

**Suggestion:** When you prepare the sql query, you should use placeholders to specify parameters. Make shure to use the right syntax, i.e, '%s' for MySQL/MariaDB and '?' for SQLite.

### **/newTask <TASK TO ADD>**

Add a new task to the "task\_list.db" database.

### **/removeTask <TASK TO REMOVE>**

Remove a task from the "task\_list.db" database by typing exactly its content.

### **/removeAllTasks <SUBSTRING TO USE TO REMOVE ALL THE TASKS THAT CONTAIN IT>**

Remove all the existing tasks from the "task\_list.db" that contain a provided.