Database Systems

by José Moreira, University of Aveiro

Homework assignment #2: database applications

Part I (60%)

Th first part consists of creating a database and importing data from the "StudentLife – Behavioral data from mobile phones" dataset, using Python and SQL. StudentLife is a study that "uses passive and automatic sensing data from the phones of a class of 48 Dartmouth students over a 10 week term to assess their mental health (e.g., depression, loneliness, stress), academic performance (grades across all their classes, term GPA and cumulative GPA) and behavioral trends (e.g., how stress, sleep, visits to the gym, etc. change in response to college workload -- i.e., assignments, midterms, finals -- as the term progresses)".

The dataset is available at https://www.kaggle.com/datasets/dartweichen/student-life. The topics that you must consider are:

a) Dinning.
b) Grades (in Education).
c) One of the datasets, e.g., PerceivedStressScale, in Survey.
d) Another dataset at your choice.

You must also create a table to hold the following data about each student: id, gender, age group, year (1 to 5) and area (social sciences, life sciences, engineering or mathematics).

The tasks to be implemented are:

1. Design and implement a data model to represent the students' data and the topics mentioned above in a Microsoft SQL Server or PostgreSQL database.
2. Create a script in SQL to simulate and load data into the students' table.
3. Create the scripts in SQL and/or Python to import the selected data into the database. You must import the data of at least 6 students.
   Tip: If data insertion directly via Python takes a long time, you can prepare Python scripts to prepare the data in a format suitable for loading into the database, if necessary, save that data in csv files, and import the data in the csv into the database.
4. Create a script to read data from the database and plot a chart of your choice using matplotlib, seaborn or another library of your choice.

You must automate the processes to create the tables and load the data into the database so that user input is minimized. For instance, you can create a configuration file providing the location and name of the files to be imported.

Part II (40%)

The second part consists of implementing two tutorials on developing web applications in Django. The first tutorial presents the basics and the second shows how to create a simple application to create, retrieve, update and delete (CRUD) data in a database.

1. Get started with the Django web framework.
   https://docs.microsoft.com/en-us/visualstudio/python/learn-django-in-visual-studio-step-01-project-and-solution?view=vs-2022

   This tutorial also shows how to store and maintain a project on github. Github is a source code management and version control system widely used by software developers around the world. This topic is optional and you do not need to implement it in this tutorial (but if you wish to do so, you must create an account in github).

Unfortunately, the Visual Studio for Mac distribution does not yet support Django projects. If you have a Mac you can install PyCharm, a well-known IDE for Python programming. The PyCharm Community Edition is available at https://www.jetbrains.com/pycharm/download/#section=mac. To create a new project in PyCharm you must perform the steps described in https://www.jetbrains.com/help/pycharm/creating-django-project.html instead of those listed in the Visual Studio tutorial.

If you wish, you can suggest and implement a tutorial of your choice similar to this one.

If you use Visual Studio you must implement steps 1 to 5 of this tutorial. If using PyCharm you should only implement steps 1 to 3.

2. Build simple CRUD app in Django.
   https://www.codershubb.com/build-simple-crud-app-in-django/

   The instructions presented in this tutorial are independent of the IDE (Visual Studio or PyCharm). Thus, it is recommended to complete the previous tutorial to acquire the basic knowledge to implement this tutorial. At the end of this document, you will find instructions and links to pages explaining how to perform in PyCharm the tasks in this tutorial that are specific to the IDE (Visual Studio or PyCharm).

3. Choose a table from your database, e.g., the students table created in Part I of this assignment or a table from the company database used in the practice problems on SQL, and create CRUD application to insert, update and delete data in that table.

   Microsoft SQL Server is not yet supported natively by Django, so it is necessary to install packages to link these tools. However, there is a known issue when using Microsoft SQL Server and newer versions of Django (3.1 and later). So, if you want to use Microsoft SQL Server, you must downgrade Django to version 3.0. Note that this can cause problems when implementing other application components because some packages may require the use of newer versions of Django. To avoid problems, it is recommended to use PostgreSQL.

Instructions

1. Each team must have two students and the teams should not be the same as in the previous homework assignment.
2. Due date: June 26.

Deliverables

You must submit the scripts and projects properly organized by part and task. If necessary, include a readme file with instructions for understanding and executing the implemented programs.

Annex – Build simple CRUD app in Django

Visual Studio

Create a Django project from the blank template and then you need to add a Django App to the project.

PyCharm

1. Run the commands in the command-line:

   $ django-admin startproject crudProject

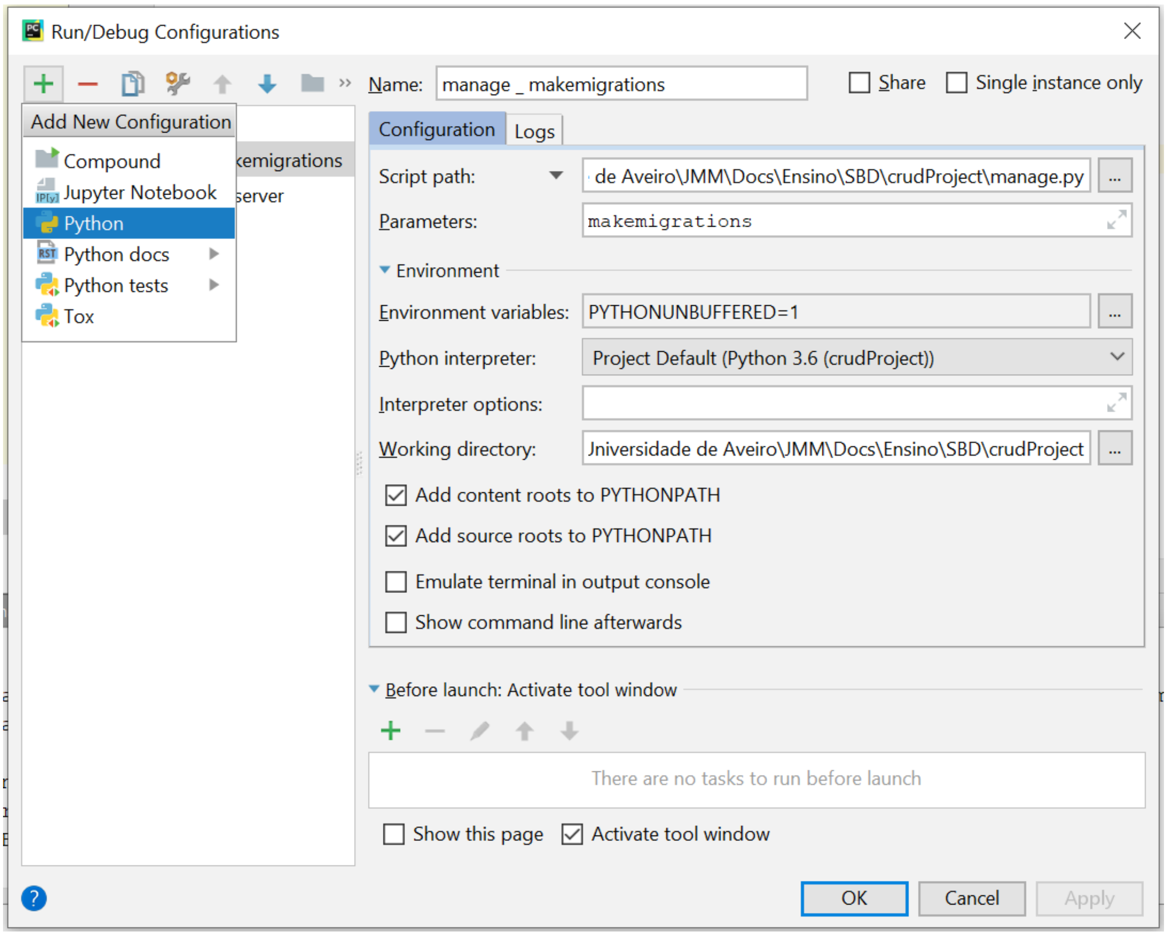
   $ cd crudProject

   $ python manage.py startapp crudApp

2. Open the crudProject in PyCharm and:

   Select the file manage.py, then execute the Run command and edit the configurations: the script path is the directory where manage.py is stored and the parameter is runserver.

   Add to more configurations on the manage.py script, one for the parameter makemigrations, as exemplified in the figure below, and the other for the parameter migrate.



   Then you just need to follow the instructions in the tutorial (edit setting.py and so on).