Databases — Part 4 (Graded Group Work)

The objective of this sequence of works is to understand the concept of databases and to handle it with the SQL language: we will manipulate this language thanks to the `sqlite3` module in Python.

1 The company example — Work that sums up what we know

In this last part of the work, we review all our knowledge on another example: a company. This company hires employees having different skills in different departments. Each employee has a computer, and some of them are also computer administrators. The big picture of the database is the following:

- departments: name, nb_machines, nb_employees, #admin_id
- computers: #dpt_name, id, os
- employees: id, surname, first_name, skill
- assignments: #dpt_name, #employee_id, year

The keys that are preceded by a # are foreign keys (linked to another table). The underlined keys are the primary keys of each table. Of course, to create the tables, it is mandatory that all tables from which we use keys are already created. Thus, we have to create first employees, then departments, then computers, and then assignments.

As in previous works, the files that contain all the necessary data to manipulate those tables are available at:
http://www.barsamian.am/2023-2024/S6ICTB/TP15_Company_creations.sql
http://www.barsamian.am/2023-2024/S6ICTB/TP15_Company_insertions.sql
http://www.barsamian.am/2023-2024/S6ICTB/TP15_Company_database.py

Remark: the departments table is not up-to-date with respect to the rest of the database, it's normal, and we will need to deal with it: the number of machines and the number of employees cannot be derived from the numbers given in this table. We will have to compute it by hand and then update the departments table.

Now, you should be able to write and test the following requests:

1. What are the departments in which there are (strictly) more than 50 employees?
2. What are the names of employees from the research department?
3. In which departments is there at least one employee whose skill is management?
4. How many employees are in the production department?
5. How many employees are there in each department? (use `GROUP BY dpt_name` instead of `WHERE ...` or write a loop in python).
6. How to update the nb_employees field of the departments table according to those values?
7. Sort by ascending order of hiring date the employees of this company.
8. How many employees have a skill that begin with "s"?
9. What are the skills that have a "g" somewhere in their name?
10. What are the departments that do not use macOs as an OS?

2 The password example — Bonus work

1. Create a new database with the following scheme:

   - people: id, surname, first_name
   - identifiers: #id, login, pwd

   (the id field in the identifiers table is a foreign key from the people table)

2. In the people table, add 5 people of your choice.

3. Write a function that, given a surname and a first_name, outputs a login. The function must insert this login inside the identifiers table for each people in the people table (the pwd is a random string that uses 4 digits).

   Hint: Don't forget to handle the case where a login already exists. For the passwords, do you have to take special care? If yes, give a solution; if not, explain why.