COMP 3090-202 Fall 2023 Project

Due 11:59pm Sunday December 10

In this project, you will be implementing a hospital database. You will use SQLite to do this project.

I **strongly recommend that you start as early as possible** on this project. It may take longer than you would expect!

1. SQLite

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. In addition, SQLite source code is in the public-domain and is free to everyone to use for any purpose.

Depending on your OS environment, SQLite server may already come bundled with your Operating System. For example, Mac OS X comes pre-installed with SQLite and can be executed using the sqlite3 command. However, you may need to install SQLite in some cases and on other OSes. As in real work, you can do some investigation online on your own to set up the environment, as part of your task of the project.

Note: Each DBMS may have its own "dialect" on certain SQL structures and usage. In case you get some error, you should explore yourself if alternative SQL statements may work out.

2. Database Design

You will implement a database to store and manage all the activities of a hospital. We would like to store the following information in a database.

There are many *doctors* in the hospital; we'd like to record their identification (ID), name, gender, age, phone number, address, years of experience and specialization. We also have *patients*, who have identifier (ID), name, gender, age, phone number, address, and disease information. Clearly, each doctor has a number of patients, and vice versa. A doctor is assisted by a number of *nurses*, for whom we record ID, name, address, specialty, and the shift during a day (we number it as two possible values 1 and 2, denoting different time slots of a day).

Doctors perform *tests* over certain patients. A *tests* table stores information of patient on whom test is performed, the doctor who performed the test, and details of the test like the date of the test, its result (positive or negative) and the instrument used for the test. Each test uses at least one of the instruments, which in turn has ID, name, and manufacturer.

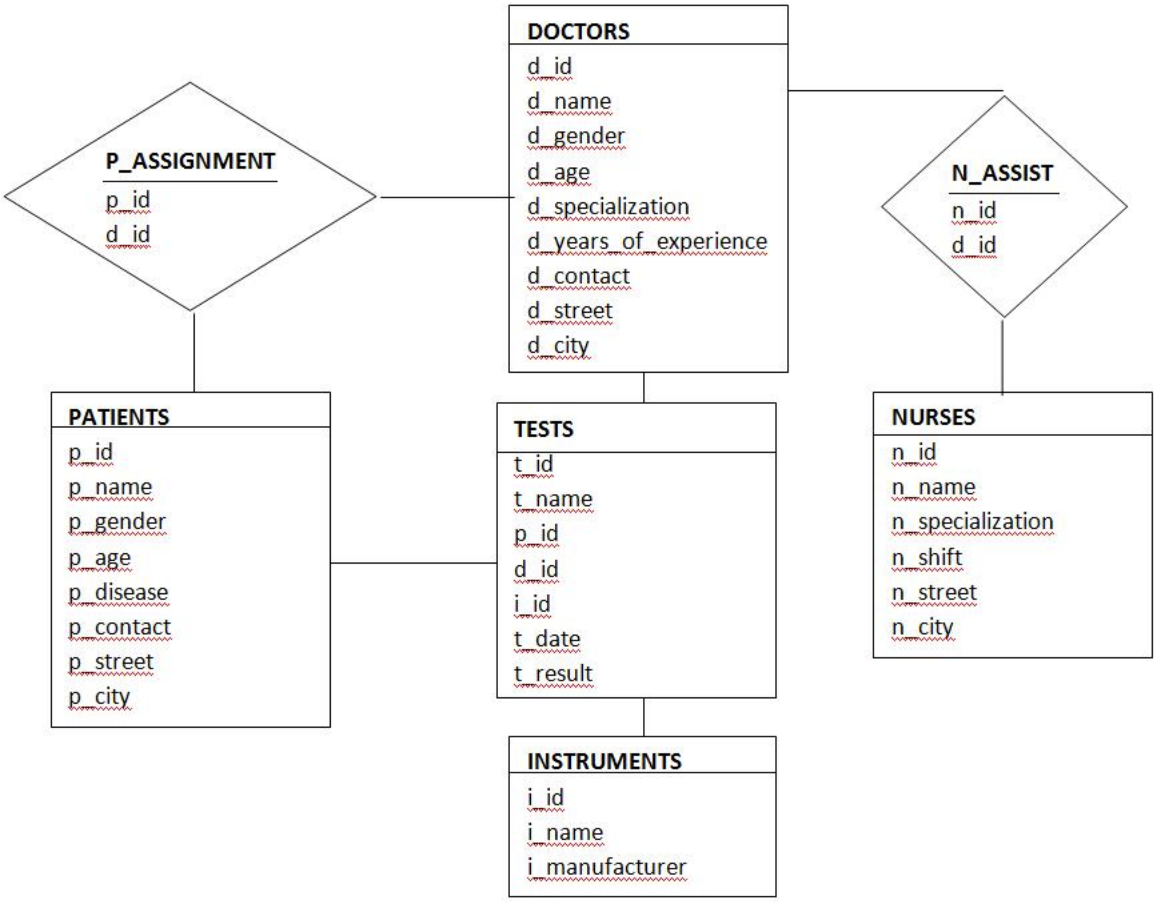

Figure 1: E-R Diagram of Hospital Database

3. What you need to do

In this project you need to implement the hospital database with tables exactly same as the provided schema. Primary keys of tables are underlined in the schema. You should import the provided dataset into the database and perform the given queries below. You should do some research online and learn about how to import a CSV file into an SQLite table. For your convenience, you are provided with a downloaded tutorial on this.

SCHEMA:

DOCTORS (D_ID, D_NAME, D_GENDER, D_AGE, D_SPECIALIZATION, D_YEARS_OF_EXPERIENCE, D_CONTACT, D_STREET, D_CITY**)**

PATIENTS (P_ID, P_NAME, P_GENDER, P_AGE, P_DISEASE, P_CONTACT, P_STREET, P_CITY**)**

NURSES (N_ID, N_NAME, N_SPECIALIZATION, N_SHIFT, N_STREET, N_CITY**)**

P_ASSIGNMENT (P_ID, D_ID**)**

N_ASSISTS (N_ID, D_ID**)**

TESTS (T_ID, T_NAME, P_ID, D_ID, I_ID, T_DATE, T_RESULT)

INSTRUMENTS (I_ID, I_NAME, I_MANUFACTURER)

QUERIES:

1) List all the doctors RICHARD MILLER is consulting.
2) Find all the test results of cancer patients. (Note: There may be different type of cancer)
3) List all the instruments produced by a manufacturer whose name starts with "S".
4) Find the most experienced doctor in the hospital.
5) List all the patients of doctor JAMES SMITH who live in the same street and same city as him.
6) Find the nurses who assist at least two doctors. Display nurse name and the number of doctors he/she is assisting
7) List the doctors and the number of nurses they have in the descending order of their number.
8) Find all the nurses who are not assigned to any doctors.
9) Increment years of experience of all the female doctors by 5.
10) Delete all the tests whose result is negative.

Finally, you must dumb the whole database into a file named “hospitaldb.sql”, using the SQLite “dump” command. Again, you can do some research online for it, but for your convenience, we provide you a downloaded tutorial on using the “dump” command.

Extra Credit: (up to 30 points beyond the total 100 points)

If you wish, you could also perform all the queries from a database access interface in an application programming language (e.g., JDBC in Java or another programming language). However, this would be some extra work if you wish to earn some extra credits. If you choose to do it, you need to submit your application program (e.g., Java program with JDBC code) along with the SQL queries executed within it. Also include readme on how to compile and run it.

4. Logistics

4.1 Collaboration

You have plenty of time and this project is manageable by a single person. Therefore, to the benefit of your learning experience (you will understand why), no collaboration is allowed in this project. You must write your own SQL queries. If needed, you could choose to discuss with another student on high-level ideas, but you must indicate whom you have discussed with in your write-up. Even so, you still must understand the project and implement by yourself. You will get virtually no grade for this project if we find you cheating.

4.2 Submission

You need to turn in following:

- **Write-up:** Should contain the following
  - Queries and results (copy-paste from the interface)
  - Describe any missing or incomplete pieces in the project.

- **The dumped database file "hospitaldb.sql"**.

Zip the files and submit it in Blackboard. For project related questions, you may also reach out to Emil Zulawnik (Emil_Zulawnik@student.uml.edu) and Jason Huang (yaoching_huang@student.uml.edu).