EECE 433 – Database Systems
Database Lab Assignment 5, Spring 2024
Finally… DB From A to Z !!!! ☺
From requirements to implementation
Duration: 3 weeks

<u>This programming assignment is due on Sunday March 31, 2024 at 11:00 pm.</u>

Assignment Objectives:

This assignment represents a critical success milestone for the course, where the objective is to have students demonstrate the ability to achieve the full cycle of DB application development: From DB requirements, to analysis, to conceptual design (ER), to logical design (Schema), to implementation (Application and queries), and finally a console and a web application.

From Requirements to Implementation

In this part of the assignment, the students are required to demonstrate the ability to take database requirements, follow the learned design process, and deliver a database implementation.

<u>Requirements:</u>

Consider the following set of requirements for a Pharmacies-Chain database
The Prescriptions-R-X chain of pharmacies has offered to give you a free lifetime supply of medicine if you design its database. Given the rising cost of health care, you agree. Heres the information that you gather:

- Patients are identified by a pid, and their names, addresses, and ages must be recorded.
- Doctors are identified by an did. For each doctor, the name, specialty, and years of experience must be recorded.
- Each drug medicine is identified by a mid, name, and formula.
- Pharmacies are identified by a hid, name and location
- A patient can be examined by many physicians (Doctors), but each patient should have one primary physician (Primary Doctor).
- Many patients can have the same primary physician.
- A doctor should have at least one patient.
- Each pharmacy sells at least one drug and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
- Doctors prescribe drugs for patients. A doctor could prescribe zero, one or more drugs for several patients, and a patient could obtain prescriptions from several doctors.
- Each prescription has a date and a description associated with it. If a doctor prescribes the same drug for the same patient more than once, all such prescriptions need to be stored.
- Pharmacies have a supervisor who is a doctor. There must always be one supervisor for each pharmacy.

Conceptual Design:

1. Design an ER schema for this application, and draw an ER diagram for the schema. Specify:
   a. Different entities and relationships.
   b. Attributes for each entity and relationship.
   c. The primary Key for each entity type.
   d. Specify entity attributes that also need to be unique.
   e. Make a note of any unspecified requirement and make appropriate assumptions to make the specification complete.
   f. Identify what can be NULL and what cannot be NULL. In your design have at least two attributes NULL

Make sure you submit documented ER diagrams with explanations along with the rest of your assignment deliverables.

Logical (Relational) Design:

2. Derive the corresponding relational database scheme. Then implement (in PostgreSQL server) the SQL statements for creating the derived relations for the entities and relationships in the ER diagram (DDL Statements). Identify all the primary and foreign key constraints in your diagram. Identify also what may and may not be NULL.

Populating the Database with meaningful data

- For each of the created relations, fill the tables with meaningful data.
- Each table entity should have at least 2 entries. Each table relationship should have at least 4.
- Make sure integrity constraints are preserved while filling the tables, i.e. no violation of the constraints is allowed.
- The function now() in PostgreSQL Server returns the current date.

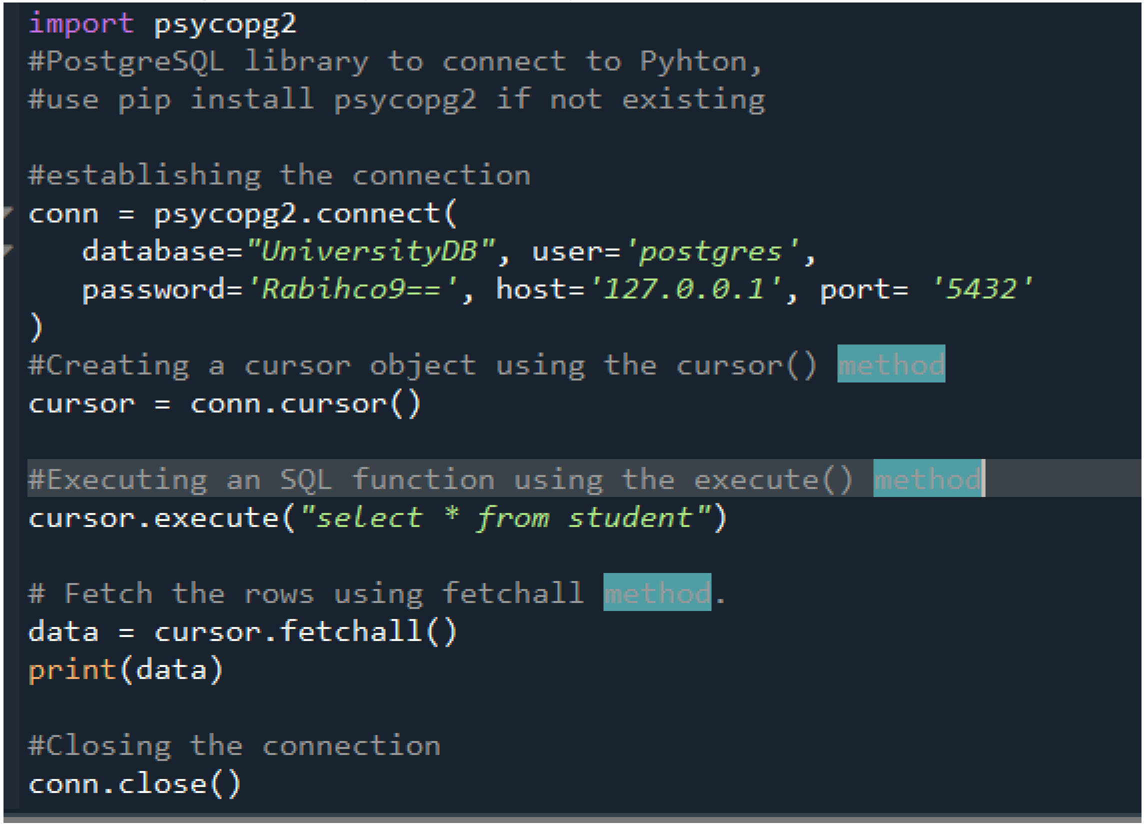
Writing the needed queries:

You will be asked in the implementations to display or manipulate data. You need to write the needed queries and add them in the Python code.

Database Implementation

DB Console Application

DB connection to PostgreSQL
Use the following code: modify the connection parameters as needed



```
import psycopg2
#PostgreSQL library to connect to Pyhton,
#use pip install psycopg2 if not existing

#establishing the connection
conn = psycopg2.connect(
   database="UniversityDB", user='postgres',
   password='Rabihco9==', host='127.0.0.1', port= '5432'
)
#Creating a cursor object using the cursor() method
cursor = conn.cursor()

#Executing an SQL function using the execute() method
cursor.execute("select * from student")

# Fetch the rows using fetchall method.
data = cursor.fetchall()
print(data)

#Closing the connection
conn.close()
```

3. Design a repetitive menu driven Python application to display the content of the database, with the following capabilities:

- Display the table Doctor in a nice tabular format.
- Display the table Patient in a nice tabular format.
- Given a doctor (did), the application will display:
    - The name of the doctor.
    - The number of patients as primary he/she has.
    - The names of his/her patients as primary.
    - The pharmacies he/she supervises if there are any.
    - ALL his/her prescribed drugs for all his/her patients
    - Issue a ‘not found’ message if doctor ID not found.
- Given a patient (pid), the application will display:
    - The name of the patient.
    - Age of the patient.
    - The name of his/her primary doctor.
    - The prescribed drugs he/she is taking.
    - Issue a ‘not found’ message if patient ID not found.

- Insert a new patient, taking into consideration:
    - No duplicate primary key, otherwise error.
    - If a field can be null, empty in input means null in database.
    - Assign directly a primary doctor to the patient, showing the doctors IDs available so that you can assign one of them.

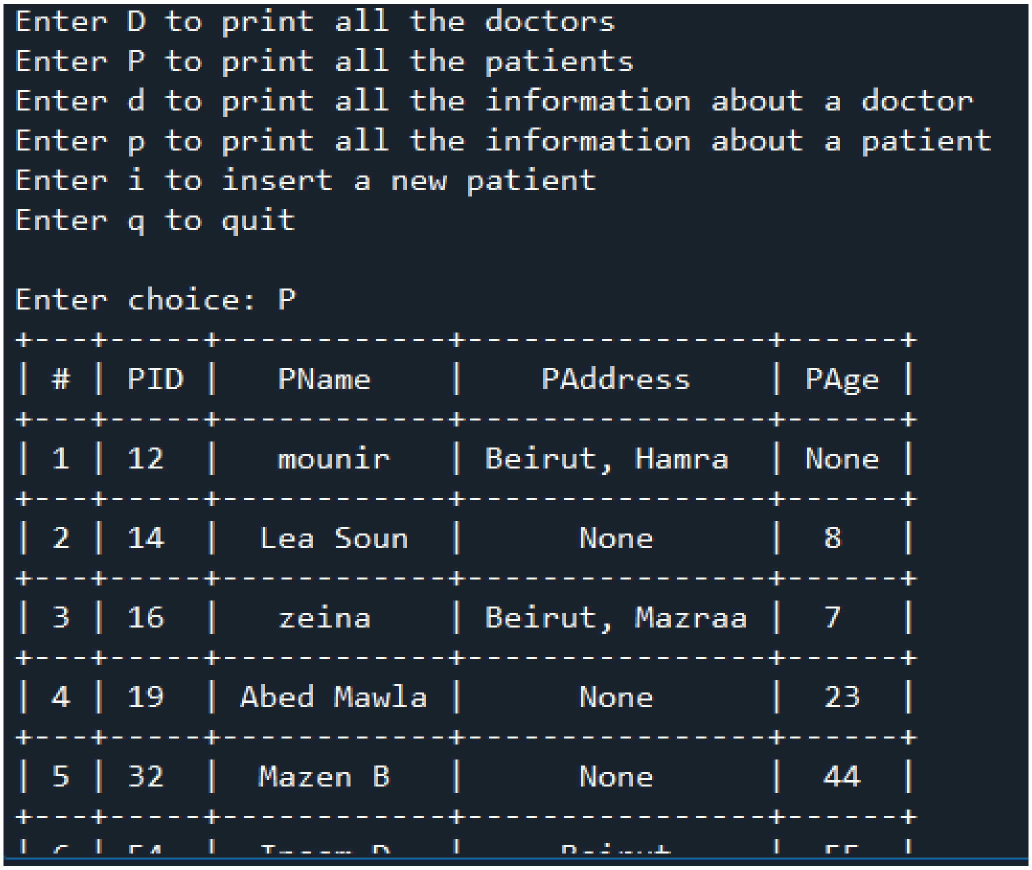
Sample run:

For the menu and display the patient table,

Here I used a tabular format called **beautifultable**, not in spyder by default.

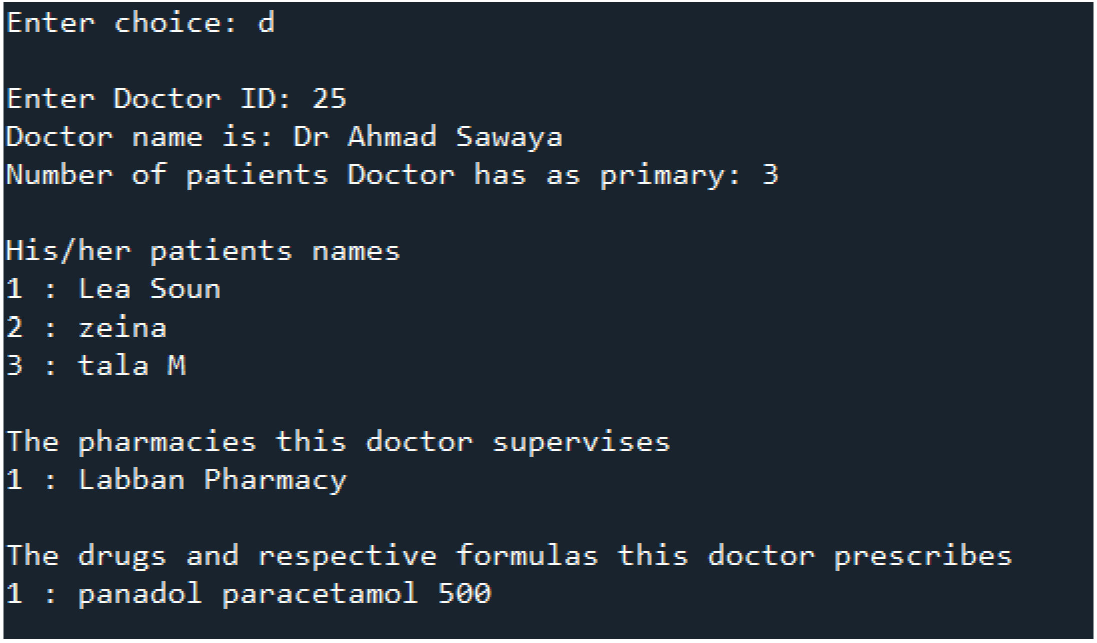
To install it use pip install **beautifultable** in anaconda prompt.

You can use other nice format, but not print a list directly.



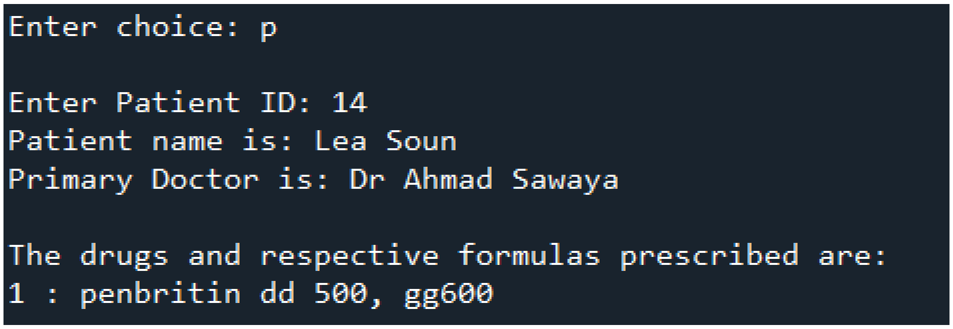
```
Enter D to print all the doctors
Enter P to print all the patients
Enter d to print all the information about a doctor
Enter p to print all the information about a patient
Enter i to insert a new patient
Enter q to quit

Enter choice: P
+---+-----+------------+----------------+------+
| # | PID |   PName    |    PAddress    | PAge |
+---+-----+------------+----------------+------+
| 1 | 12  |   mounir   | Beirut, Hamra  | None |
+---+-----+------------+----------------+------+
| 2 | 14  |  Lea Soun  |      None      |  8   |
+---+-----+------------+----------------+------+
| 3 | 16  |   zeina    | Beirut, Mazraa |  7   |
+---+-----+------------+----------------+------+
| 4 | 19  | Abed Mawla |      None      |  23  |
+---+-----+------------+----------------+------+
| 5 | 32  |  Mazen B   |      None      |  44  |
+---+-----+------------+----------------+------+
```

Displaying the information about a doctor



```
Enter choice: d

Enter Doctor ID: 25
Doctor name is: Dr Ahmad Sawaya
Number of patients Doctor has as primary: 3

His/her patients names
1 : Lea Soun
2 : zeina
3 : tala M

The pharmacies this doctor supervises
1 : Labban Pharmacy

The drugs and respective formulas this doctor prescribes
1 : panadol paracetamol 500
```

Displaying the information about a patient:

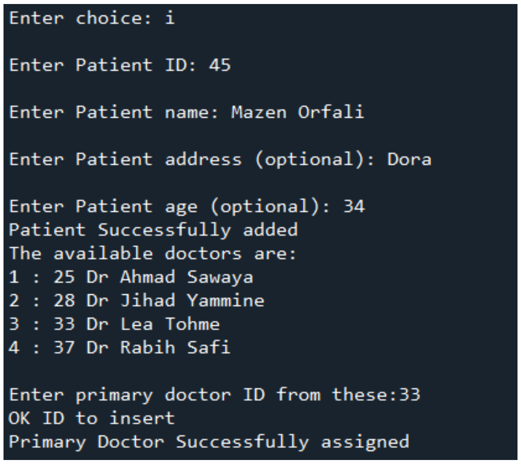


```
Enter choice: p

Enter Patient ID: 14
Patient name is: Lea Soun
Primary Doctor is: Dr Ahmad Sawaya

The drugs and respective formulas prescribed are:
1 : penbritin dd 500, gg600
```

Inserting a new patient:

Note the retrieval of the Doctor IDs and names so that to select one of them

The optional fields can be empty → converted to null in database



```
Enter choice: i

Enter Patient ID: 45

Enter Patient name: Mazen Orfali

Enter Patient address (optional): Dora

Enter Patient age (optional): 34
Patient Successfully added
The available doctors are:
1 : 25 Dr Ahmad Sawaya
2 : 28 Dr Jihad Yammine
3 : 33 Dr Lea Tohme
4 : 37 Dr Rabih Safi

Enter primary doctor ID from these:33
OK ID to insert
Primary Doctor Successfully assigned
```

DB Web Application

The lab instructor will perform a short tutorial on a Web application including:

- Some HTML knowledge.
- A look on the index.html file.
- Python Web library: flask.

Tutorial Description:

One table database, to be done in the Lab with instructor help

Step 1: Create a database

- Using **PostgreSQL** create a database then a student table with sid, sname, sgrade, and sdob. Make sid integer, sname char(20), sgrade real, and sdob date type. Only sdob can be null.
- Insert at least six students. Use meaningful values.
  To connect to **PostgreSQL** , use the connection string we did for the console application:
- Compute and store in a variable the largest grade.
- Compute and store in a variable the student who got the largest grade.
- Compute and store in a variable the grade average.
- Compute and store in a variable the student year of birth.
- Remember these variable names, you will need them later on when rendering your values on the browser.
- Run and test your code.
- Save your file so that you can reuse it below.

Step 2: Basic DB Web application using Python with SQL query

A. Understanding Index file

- Get familiar with the index.html file. For this assignment, we have provided you with the sample index.html provided.
  - Its location must be in a logical subdirectory called templates. If your python file is in the desktop, the index.html file must be in desktop/templates.
  - Make a copy of the index file and save a backup copy. You will need it for the parts below.
  - Make another copy with comments to explain what do the different items signify, including table attributes (td, tr …), the tags (<h1> <h3> <h4>) , and the braces which are values rendered from python flask.

B. Generate and display the data expected by the index.html file

- Install the web python library **flask.**
- Update the python code to add support for web rendering:

```
app = Flask(__name__)
@app.route("/")
def main():
   return query_mysql('SELECT * from Student')
if __name__ == "__main__":
   app.run()
```

- Note that query_mysql is your active function and its parameter is the SQL query that selects the three fields.

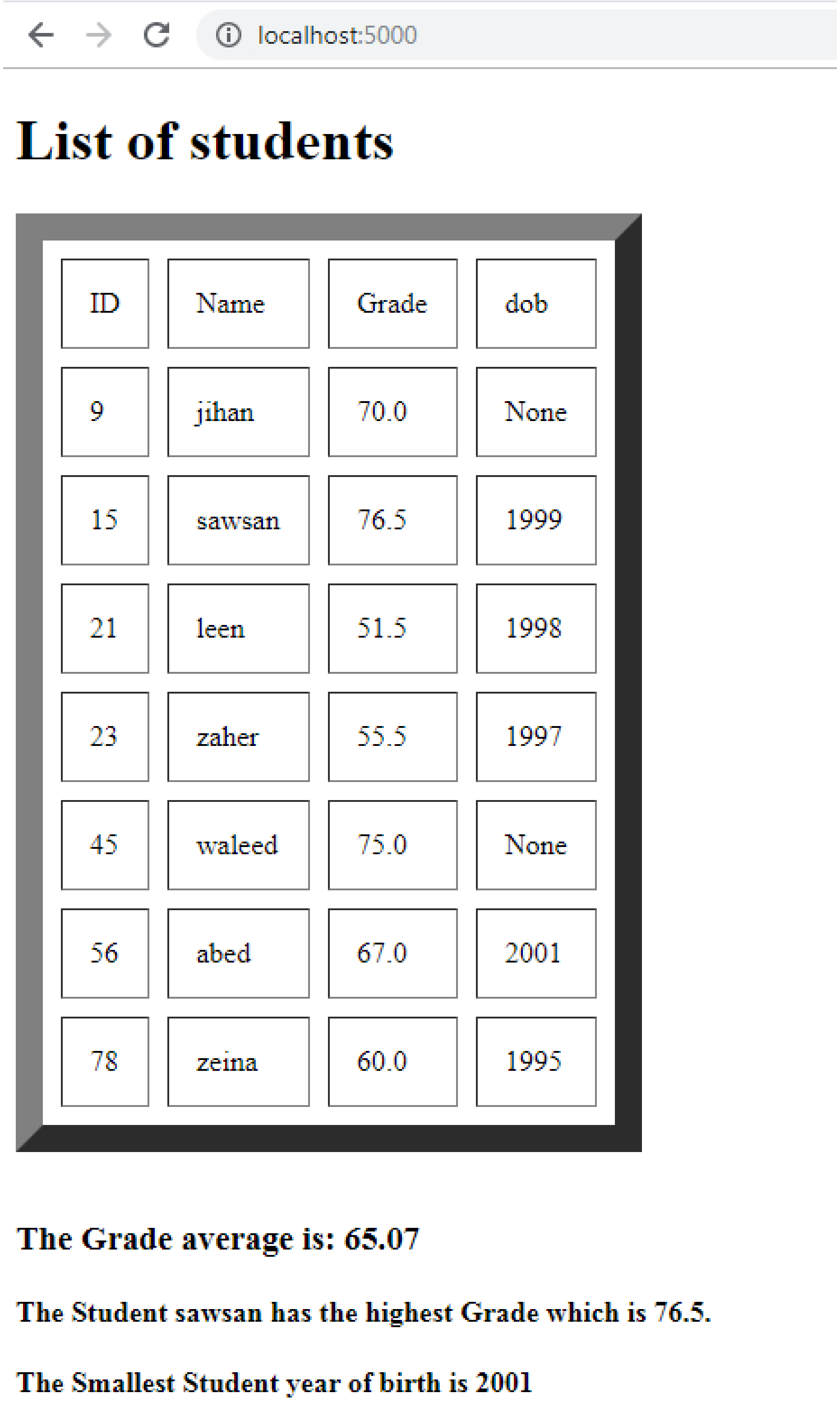
- Modify the index.html so that it only renders the data available in this part.
- Run the code and report then open the browser and type localhost: 5000. 5000, which is the port number used by flask. Report on results (see below for sample results).

Step 3: Data manipulation and web data display using flask

- Use the computations you did in step 1 to return the values to the default browser by writing something like:

  return render_template("index.html", rows = rows, cc = val, average=avg)

  The variable names indicate the python variable, as used in step 1 above, and the tagged variable in the index.html file. Left one is the variable name in the html file, right one is the variable name in the python file. Note that it can be the same if it is clearer and easier for you.
- render_template will execute on the server, the parameters are the values you computed in step 1. These will be displayed on the web page.
- Use the index.html provided to you for this assignment to display the data needed here.
- Run the code and report on browser results. (see below for sample results).



List of students

ID	Name	Grade	dob
9	jihan	70.0	None
15	sawsan	76.5	1999
21	leen	51.5	1998
23	zaher	55.5	1997
45	waleed	75.0	None
56	abed	67.0	2001
78	zeina	60.0	1995

The Grade average is: 65.07

The Student sawsan has the highest Grade which is 76.5.

The Smallest Student year of birth is 2001

Note that date can be null, it will be displayed None. Also we are only interested in the year of dob in the display and computation.

Step 5: Data Entry through Web

You need to add an html form, with 3 input boxes for the 3 values to insert, as dob can be null. labels for display, and a button for save. Make a new copy of the original index.html file, add the code to allow inputting a student row of values (ID, name grade) and insert them in the table.

- Redisplay all the table after successful insertion, otherwise display an error message.
- Add a fourth input Box to accept a date. This box can be filled with a valid date or remain empty when inserting.

This part has to be submitted and will be graded, together with the console application

We want to update the previous part, so that instead of having a console application, we want to have a web application, and display the information of the pharmacy database on the browser.

4. DB Web Application

Design a Python Flask web application to display the content of the database, with the following capabilities:

A-

- Display all the entity tables data.
- Display the total number of patients.
- Given a doctor (did), the application will display:
  - The name of the doctor.
  - The number of patients he/she has.
  - The names of his/her patients.
  - The pharmacies he/she supervises if there are any.
  - ALL his/her prescribed drugs for all his/her patients.

B- Insert a new patient and assign to him/her a primary doctor.

C- Insert a new doctor and make him/her either a normal or primary for a specific patient.

D- Insert a new pharmacy and assign to it a supervisor(doctor).

E- Create a new valid prescription.

F- Input a location, display the patients living there and the existing pharmacies if any, you need to use SQL like with %.

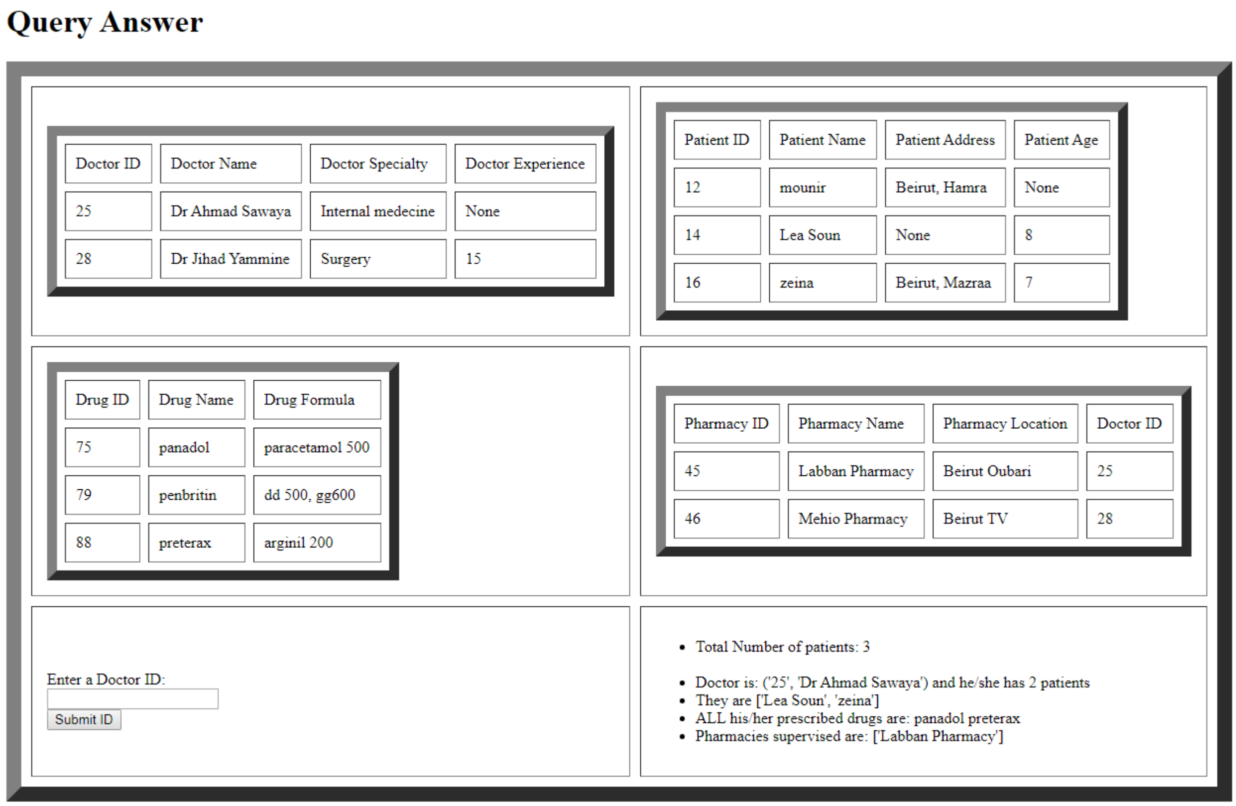
G- Display all drugs assigned together with the number of doctors who prescribed each.

H- Input a patient name, the application will display:

a. The patient ID.
b. The patient primary and other doctors.
c. The patient drugs taken so far, if any.

Make sure your insertions do not violate any constraint. If yes indicate through a message the kind of violation: Primary or Foreign keys, empty for non-null fields, format violation…
It is very important to avoid any crash.

Sample Run for the above part A: (a html file template is provided: index4T)



Query Answer

Doctor ID	Doctor Name	Doctor Specialty	Doctor Experience
25	Dr Ahmad Sawaya	Internal medecine	None
28	Dr Jihad Yammine	Surgery	15

Patient ID	Patient Name	Patient Address	Patient Age
12	mounir	Beirut, Hamra	None
14	Lea Soun	None	8
16	zeina	Beirut, Mazraa	7

Drug ID	Drug Name	Drug Formula
75	panadol	paracetamol 500
79	penbritin	dd 500, gg600
88	preterax	arginil 200

Pharmacy ID	Pharmacy Name	Pharmacy Location	Doctor ID
45	Labban Pharmacy	Beirut Oubari	25
46	Mehio Pharmacy	Beirut TV	28

Enter a Doctor ID:

- Total Number of patients: 3

- Doctor is: ('25', 'Dr Ahmad Sawaya') and he/she has 2 patients
- They are ['Lea Soun', 'zeina']
- ALL his/her prescribed drugs are: panadol preterax
- Pharmacies supervised are: ['Labban Pharmacy']

Notes:

1. You may use html lists and tables for nice value display.
2. You may display results with or without quotes.
3. You need to create several web pages, find how.
4. You need to add buttons or menus to navigate through your pages and features.
5. You may add images and advanced hyperlinks.

You are free to design your web page(s) as you like. Make use of part A and the provided index4.html. Note that 90% of the grade will be on the above A to H requirements, and 10% will be on the quality of the UI design.

You may need to present this assignment part to the instructor.

What you need to submit:

- **Conceptual Design:** The answers to the questions for conceptual design. Make sure your ER Diagram is included as electronically (not by hand) drawn softcopy. This needs to be typed. Feel use Microsoft Power Point, Word, or Vizio for your drawing. You can also use on-line software like draw.io.
- **Logical Design:** The answers to the questions for logical design. Make sure your schema diagram is included as a print-out of the schema.
- **SQL Queries:** The create table queries to support the application requirement.
- **Data**: Soft copy of the **data entered** in the tables using insert into or screenshots of the data if entered manually.
- **The Python files of your program.**
- **The HTML files of the web part.**
- **Screenshots of the running like the above ones**.
- **Finally you may need to present your work directly to the lab instructor**.