Life Link

INSTALATION / USER MANUAL

HOSPITAL MANAGEMENT DATA BASE

Index

Programs:

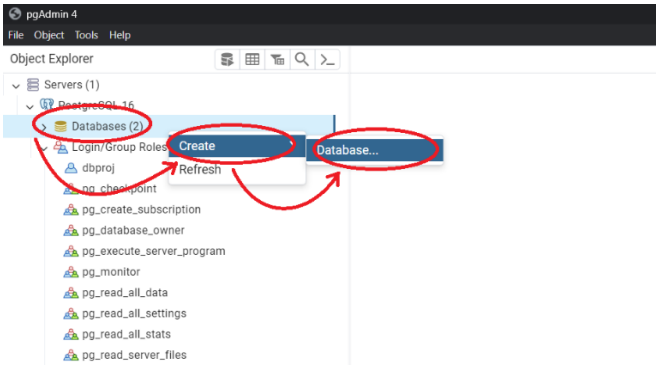
This data base is built upon the PgAdmin4 platform for its data management and storage needs, while also making use of code writen in Python. An API app of some kind is also required for the operation of the DB.

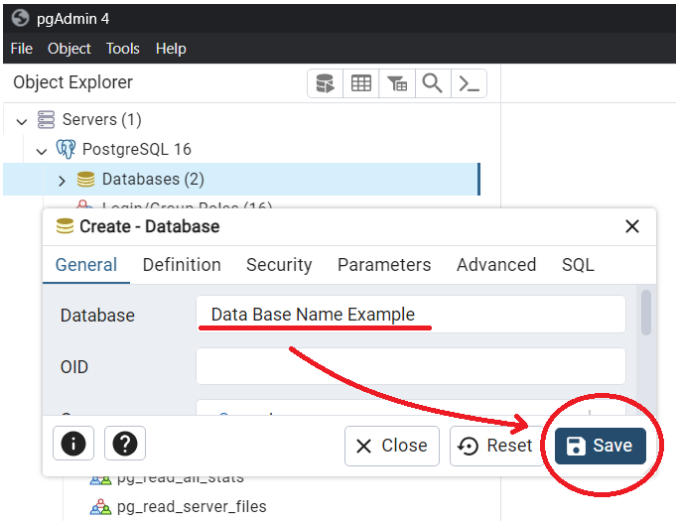
Creating the DataBase:

Once PgAdmin4 has been installed, follow these steps to create the database:

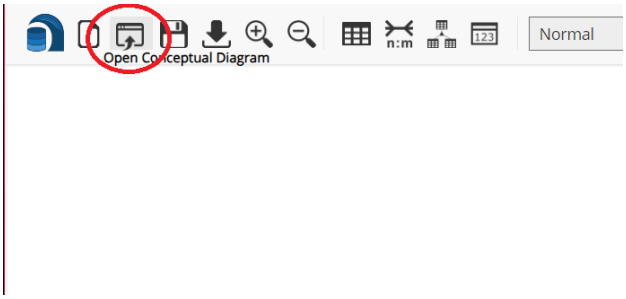
1 – Create the database

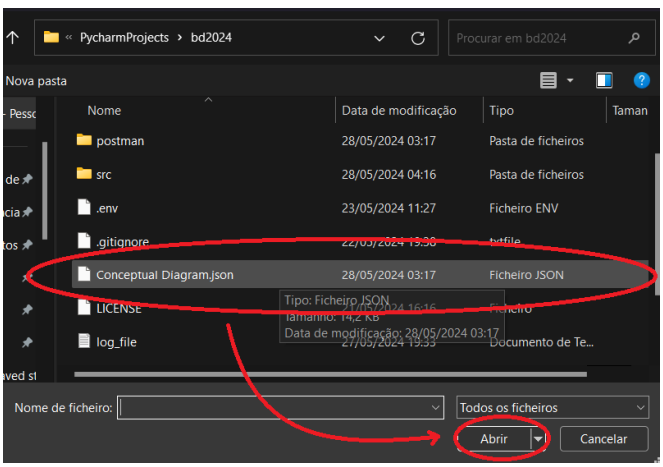
Create a new database, name it as you wish and save it.



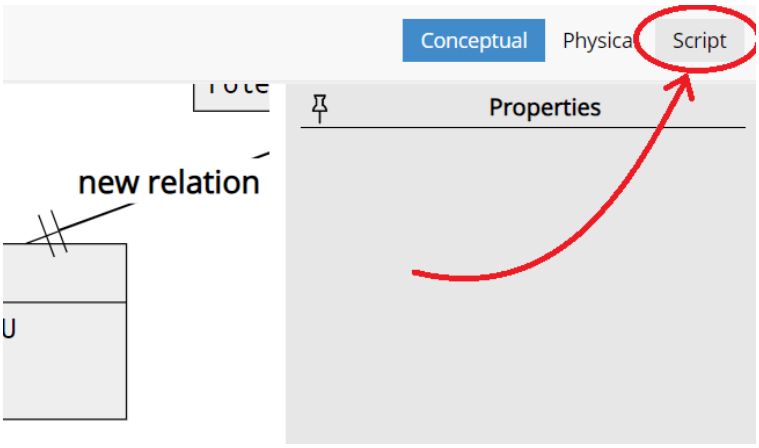


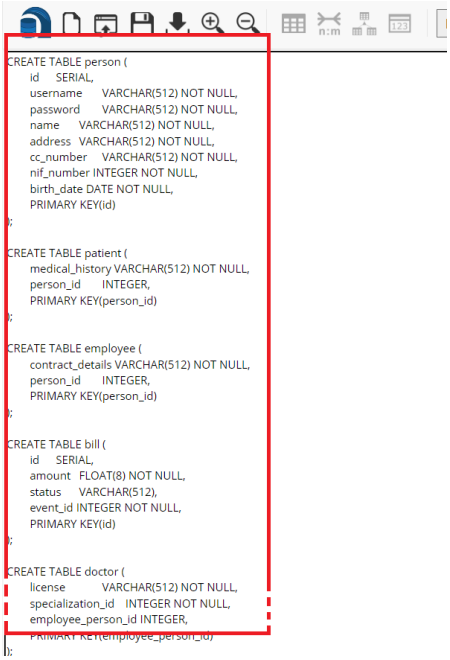
Next you'll need to create the tables and triggers. For that we'll use the Concept Diagram.json, located in the program files. Open it through Onda, an online database architect. In the website select the option in the top left corner that, when hovered over, reads "Open Conceptual Diagram" and select the json mentioned earlier from the folder located in your computer.



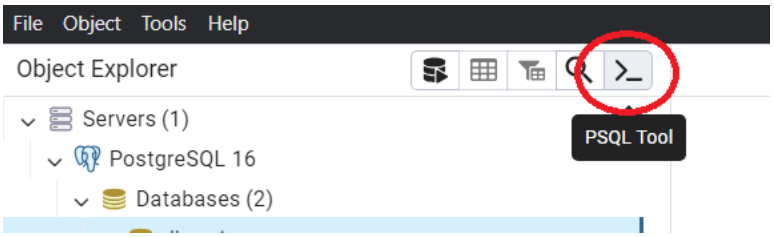


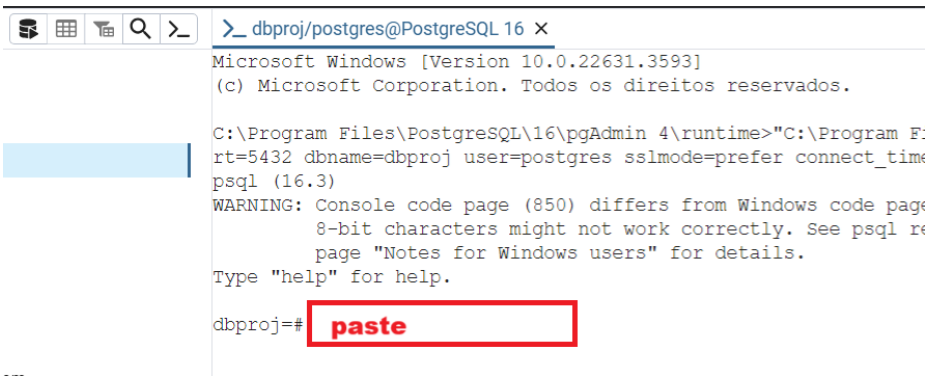
Next, on the top right corner, select the “Script” option, copy the text on the left. Then, over at PgAdmin4, paste it on the database’s PSQL Tool.



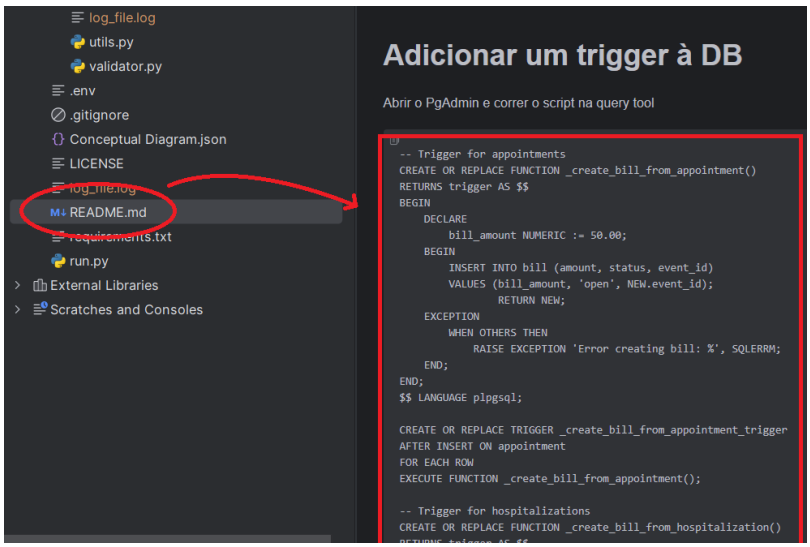


```
CREATE TABLE person (
    id    SERIAL,
    username    VARCHAR(512) NOT NULL,
    password    VARCHAR(512) NOT NULL,
    name    VARCHAR(512) NOT NULL,
    address  VARCHAR(512) NOT NULL,
    cc_number    VARCHAR(512) NOT NULL,
    nif_number INTEGER NOT NULL,
    birth_date DATE NOT NULL,
    PRIMARY KEY(id)
);

CREATE TABLE patient (
    medical_history VARCHAR(512) NOT NULL,
    person_id    INTEGER,
    PRIMARY KEY(person_id)
);

CREATE TABLE employee (
    contract_details VARCHAR(512) NOT NULL,
    person_id    INTEGER,
    PRIMARY KEY(person_id)
);

CREATE TABLE bill (
    id    SERIAL,
    amount  FLOAT(8) NOT NULL,
    status    VARCHAR(512),
    event_id INTEGER NOT NULL,
    PRIMARY KEY(id)
);

CREATE TABLE doctor (
    license        VARCHAR(512) NOT NULL,
    specialization_id  INTEGER NOT NULL,
    employee_person_id INTEGER,
    PRIMARY KEY(employee_person_id)
);
```





Then, copy the script located at the end of the README file in the project folder and paste in the database’s PSQL tool, like before.



Using the Database:

After introducing the necessary data for the DB's normal operation (medications, severities, side-effects, specializations) you are ready to start using the program. Here are the endpoints available and a brief explanation on what they do and the information they require:

(Nota: campos ocupados por Time Stamps têm que star no formado YYYY-MM-DD H:H)

Register – For a user to be able to use Life Link they need to have an account. For that purpose the User Registration endpoint is used. This platform handles two main types of users: patients and employees. The employees can be nurses, doctors and assistants. To create an account several fields need to be filled at different parts of the endpoint body. This endpoint will have additional fields for each specific type of user, however, there are several common ones. For the registration to be valid we check for any overlapping data, whether that be the username, citizen card number or taxpayer identification number. Any user can utilize this endpoint. If the registration is valid the status 200 will be returned.

Common fields: username (VChar), password (VChar), name (VChar), address (VChar), cc_number (VChar), nif_number (VChar), birth_date (TimeStamp)

Extra, user specific fields:

patient – medical history (VChar)

doctor – contract details (VChar), license (VChar), specialization name (VChar)

nurse – contract details (VChar)

assistant - contract details (VChar), certification details (VChar)

Authenticate user – After the registration, the user can use the Life Link platform through the User Authentication endpoint and should only insert their username and password on their respecting fields. If the authentication is a success the token associated to that account will be returned. After this step, the received token will be used to access the following endpoints, this will be inserted into the field Authorisation from the Header of the endpoint. Any user can utilize this endpoint.

Fields: username (VChar), password (VChar)

Schedule appointment – This endpoint can only be used by patients. The information needed is the doctor ID, the date the appointment will be scheduled to and the nurse/nurses that will be needed. In this case, to check for a valid appointment, if the doctor ID or any of the nurses aren't registered and if the date is set to a past date.

Header: token

Fields: doctor id (VChar), date (Time Stamp), nurses (list of lists, with two VChar's each)

Get appointments – This endpoint can only be used by assistants and the targeted patient. The only information that needs to be added is the patient ID, if the credentials don't match any patient it will return the status 500. Otherwise, it will return a list of dictionaries with all the prescriptions of the targeted patient.

Header: token

Schedule surgery – This endpoint is exclusive to assistants. There are two types of posts that can be done here, one where the surgery is associated with an hospitalisation, and when it isn't. In the case there isn't any hospitalisations, the information needed is the patient ID, the doctor ID, the date, the nurse/ nurses ID and role, the duration of the hospitalisation and the head nurses ID. If the surgery has an hospitalisation associated the information needed is the patient ID, the doctor ID, the nurse/ nurses ID and role, and the date. This endpoint will return all the details of the surgeryHeader: token

Fields: patient id (Integer), doctor id (Integer), date (Time Stamp), nurses (list of lists, with two VChar's each), hospitalization duration (Integer), hospitalization's head nurse id (Integer)

Add prescriptions - In case a doctor wants to add a prescription, they should use the Add Prescription. For this action to be valid we check the name of the medicine, severity and side effects. In case of success, this will return the status 200. Only doctors can use this endpoint.

Header: token

Fields: type (VChar), event ID (VChar), validity (VChar), medicines (a listo f elemtns containing medicine (VChar), posology dose (VChar) and posology frequency (VChar))

Get prescriptions – This endpoint can be used by all employees and the targeted patient. The only information that needs to be added is the patient ID, if the credentials don't match any patient it will return the status 500. Otherwise, it will return a list of dictionaries with all the prescriptions of the targeted patient.Header: token

Execute payment – This endpoint is exclusive to patients, since they are the only ones who can pay their own bills. To pay the bill the only information that needs to be given is the bill ID, the amount to be payed and the payment method. Once again, if the bill ID doesn't match with any bill in the data base it will return status 500. Otherwise, it will return the amount that is left to pay from that bill.

Header: token

Fields: amount (VChar), payment method (VChar)

List top 3 patients – Being available to any user, this endpoint's purpose is to return the three patients who've spent the most money in the current month, more especifically their names, respective amount spent and the procedures paid. Beacause of its accessibility it no credentials nor field information is required.

Daily summary – This endpoint is used to retrieve a list of information relative to all hospitalizations of a given day, such as surgeries, payments and prescriptions. Only assistants can use this endpoint, wich is verified through the provided token.

Header: token

Generate monthly report – Only accessible by assistants, this endpoint is used to retrieve a listo f the doctors who had the most surgeries each month, for the last 12 months. The data returned is a list of the name, month and count of surgeries for each month. If for some reason there are no surgeries for a given month, it is not represented in the returned list altogether.

Header: token