

1. Создание базы данных в SQLite

Загрузка ipython-sql library; % и %% - это предустановленные "magic functions"

```
In [1]: %load_ext sql

In [4]: %sql sqlite:///Patients.db

In [3]: %%sql
CREATE TABLE Aberrations(
    Patient_id INTEGER PRIMARY KEY AUTOINCREMENT,
    Journal_id INT,
    Abb_type VARCHAR(30),
    Abb VARCHAR(100)
);

* sqlite:///New.db
Done.
Out[3]: []

In [ ]: %%sql
INSERT INTO Aberrations (Journal_id, Abb_type, Abb)
VALUES
(125, "Дупликация", "Xq28(154,019,722-154,032,369)×3"),
(186, "Делеция", "6q11.1q14.1(62,118,669-80,897,321)×1"),
(200, "Делеция", "1p32.1p31.1(59,525,570-71,600,905)×1"),
(205, "Дупликация", "7q21.2(91,490,967-91,550,018)×3"),
(300, "Делеция", "Yq11.23(26,584,213-27,420,219)×0"),
(407, "Дупликация", "Xp22.33(614,934-631,408)×3");

In [ ]: %%sql
select * from Aberrations

In [ ]: %%sql
CREATE TABLE Age(
    Patient_id INTEGER PRIMARY KEY AUTOINCREMENT,
    Journal_id INT,
    Age INT
);

In [ ]: %%sql
INSERT INTO Age (Journal_id, Age)
VALUES
(125, 1),
(186, 4 ),
(200, 1),
(205, 9),
(300, 16),
(407, 2);

In [ ]: %%sql
select * from age

In [ ]: %%sql
CREATE TABLE Gender(
    Patient_id INTEGER PRIMARY KEY AUTOINCREMENT,
    Journal_id INT,
    Gender VARCHAR(30)
);

In [ ]: %%sql
INSERT INTO Gender (Journal_id, Gender)
VALUES
(125, "Женский"),
(186, "Женский"),
(200, "Мужской"),
(205, "Мужской"),
(300, "Мужской"),
(407, "Мужской");

In [ ]: %%sql
select * from Gender

In [ ]: %%sql
CREATE TABLE Analysis_date(
    Patient_id INTEGER PRIMARY KEY AUTOINCREMENT,
    Journal_id INT,
    Sample_received DATE NOT NULL,
    Result_ready DATE NOT NULL
);

In [ ]: %%sql
INSERT INTO Analysis_date (Journal_id, Sample_received, Result_ready)
VALUES
(125, '2015-05-23','2015-07-30'),
(186, '2015-08-13','2015-11-30'),
(200, '2016-03-11','2016-07-01'),
(205, '2018-10-03','2018-12-10'),
(300, '2019-01-07','2019-02-29'),
(407, '2019-02-23','2019-04-02');

In [ ]: %%sql
select * from Analysis_date
```

2. Импорт библиотек, необходимых для создания соединения с SQLite и реализации интерфейса

```
In [ ]: import sqlite3 #это DB-API модуль
import PySimpleGUI as sg #Это библиотека для создания интерфейса
```

3. Реализация связки sql и интерфейса; интерфейс

```
In [ ]: sg.theme('DarkAmber')    #Цвет интерфейса

#расчет количества дней
select = f"SELECT Dt.Journal_id, JULIANDAY(Dt.Result_ready), JULIANDAY(Dt.Sample_received) \
FROM Analysis_date Dt"
conn = sqlite3.connect('/Users/maria/Downloads/Patients.db')
days = {}
cursor = conn.execute(select)
for row in cursor:
    days[row[0]] = round(row[1] - row[2])

#тело программы
def main():
    # заполнение первого окна
    layout = [ [sg.Text('Данное приложение предназначено для упрощения работы с данными')],
                [sg.Text('Введите хромосому'), sg.InputText()],
                [sg.Text('Введите возраст пациента для поиска по таблице'), sg.InputText()],
                [sg.Button('Найти'), sg.Button('Отмена')]] ]

    #создаем окно
    window = sg.Window('Программа управления базой данных пациентов', layout)

    #выбор параметров из окна
    chromosome = str()
    max_age = str()
    while True:
        event, values = window.read()
        if event == sg.WIN_CLOSED or event == 'Отмена': # if user closes window or clicks cancel
            break
        print('You entered ', values[0], values[1])
        chromosome = values[0]
        max_age = values[1]
        window.close()

    #формирование запроса для sql на основе введенных пользователем данных
    if max_age == '':
        age_select = ''
    else:
        age_select = f'and A.Age = {max_age}'

    select = f"SELECT A.Journal_id, A.Age, Ab.Abb, Ab.Abb_type, Dt.Sample_received, Dt.Result_ready \
FROM Age A \
LEFT OUTER JOIN Aberrations Ab \
ON A.Patient_id = Ab.Patient_id \
LEFT OUTER JOIN Analysis_date Dt \
ON A.Patient_id = Dt.Patient_id \
WHERE Ab.Abb LIKE '{chromosome}%' " + age_select + " GROUP BY A.Journal_id"

    print(select)

    #забор данных из базы
    result = str()
    patients = []
    cursor = conn.execute(select)
    print(type(patients))
    for row in cursor:
        patients.append(row[0])
        result = result + f"Номер пациента в журнале: {row[0]}, \n Возраст: {row[1]},\n Хромосомная перестройка: {row[2]}"

    #окно вывода
    layout = [ [sg.Text('Результат вывода:'),
                [sg.Text(result)],
                [sg.Button('Узнать, сколько по времени готовился анализ', key="open"), sg.Button('Выполнить новый анализ')]] ]

    window = sg.Window('Программа управления базой данных пациентов', layout)

    while True:
        event, values = window.read()
        if event == sg.WIN_CLOSED or event == 'break':
            break

        elif event == 'open':
            output = str()
            for i in range(0, len(patients)):
                output = output + f"\n Для пациента {patients[i]} анализ готовился {days[patients[i]]} дней"
            sg.popup("Время изготовления анализа", output)
        else:
            main()
    window.close()
    return()

main()

conn.close()
```