

Задание 1

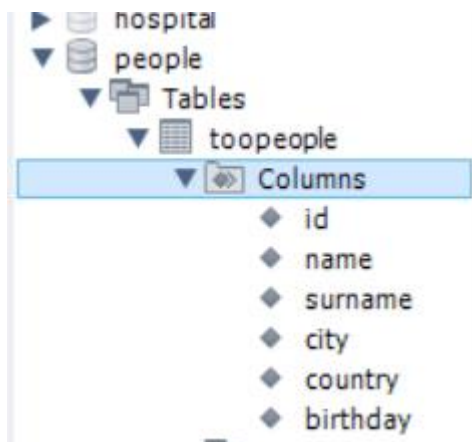
Создайте однотабличную базу данных People (имя, фамилия, город, страна, дата рождения) с одноименной таблицей. Напишите программу, которая позволяет пользователю ввести запрос и получить результаты работы запроса. В качестве запроса необходимо поддерживать только SELECT. При попытке выполнить другие запросы необходимо генерировать ошибку.

```
try:
    # Create DATABASE
    with connection.cursor() as cursor:
        create_table = "CREATE DATABASE 'people'"
        cursor.execute(create_table)

    # Create table
    with connection.cursor() as cursor:
        create_table = "CREATE TABLE 'toopeople' (id int AUTO INCREMENT, name varchar(30), surname varchar(30), city varchar(30), country varchar(30), birthday date)"
        cursor.execute(create_table)
        print("well done")

    # Insert data
    with connection.cursor() as cursor:
        insert = "INSERT INTO 'toopeople' (name, surname, city, country, birthday) VALUES ('Bober', 'Boberovich', 'Bobersk', 'Russia', '2000-01-01')"
        cursor.execute(insert)
        connection.commit()

    # Select data
    with connection.cursor() as cursor:
        select_all = input("Enter select: ")
        if "select" not in select_all:
            print("error")
        else:
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
```



Задание 2

Добавьте к первому заданию возможность вставки, удаления и обновления данных с помощью запросов INSERT, DELETE, UPDATE. Перед исполнением запроса проверяйте корректность названия таблицы. Также запретите запрос на удаление всех строк и обновление всех строк (UPDATE и DELETE без условий).

```

while True:
    v=input("Enter what you want\n1.(INSERT)\n2.(Update)\n3.(Delete)\n4.(Show all)\n5.(Show all name)\n")
    if v=="quit":
        break
    if v=="1":
        with connection.cursor() as cursor:
            insert = input("Enter INSERT: ")
            cursor.execute(insert)
            connection.commit()
    if v=="2":
        with connection.cursor() as cursor:
            create_table =input("Enter Update: ")#"UPDATE `toopeople` SET password = 'qwerty' WHERE id
            for i in create_table:
                if "*" in i:
                    print("Not Yet")
                cursor.execute(create_table)
                connection.commit()
    if v=="3":
        with connection.cursor() as cursor:
            insert = input("Enter Delete: ")#"DELETE FROM `students` WHERE name='Sasha';"
            for i in create_table:
                if "*" in i:
                    print("Not Yet")
                cursor.execute(create_table)
                connection.commit()

```

Задание 3

Модифицируйте первое задание, чтобы пользователь не мог вводить запрос, а пользовался готовыми фильтрами. Например: отображение всех людей, отображение людей с возрастом в диапазоне, отображение всех людей из одного города (задаётся в качестве значения пользователем с клавиатуры), отображение всех людей из одной страны (задаётся в качестве параметра пользователем с клавиатуры).

```

        json.dump(row, file)
        file.close()
    if v=="5":
        with connection.cursor() as cursor:
            select_all = "select name from people.toopeople"
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

    if v=="6":
        with connection.cursor() as cursor:
            select_all = "select birthday from people.toopeople"
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="7":

```

```

        file.close()
    if v=="7":
        with connection.cursor() as cursor:
            a=input("Enter city: ")
            select_all = f"select name from people.toopeople where city = '{a}'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="8":
        with connection.cursor() as cursor:
            a = input("Enter country: ")
            select_all = f"select name from people.toopeople where country = '{a}'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
while True:

```

Задание 4

Модифицируйте третье задание, чтобы фильтр для показа мог быть комплексным. Например, пользователь может указать фильтр на страну и город, после чего отобразятся люди, для которых сработает этот комплексный фильтр. Нужно поддержать условия И, ИЛИ.

```

        file.close()
    if v=="7":
        with connection.cursor() as cursor:
            a=input("Enter city: ")
            select_all = f"select name from people.toopeople where city = '{a}'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()
    if v=="8":
        with connection.cursor() as cursor:
            a = input("Enter country: ")
            select_all = f"select name from people.toopeople where country = '{a}'"
            cursor.execute(select_all)
            result = cursor.fetchall()
            for row in result:
                print(row)
                file = open(os.path.join("based", "camber.json"), "a")
                json.dump(row, file)
                file.close()

```

Задание 5

Добавьте к четвертому заданию возможность вставки, удаления и обновления данных через интерфейс приложения. Пользователь не может ввести запросы INSERT, UPDATE, DELETE напрямую.

```

while True:
    v = input("Enter what you want\n1.(INSERT)\n2.(Update)\n3.(Delete)\n")
    if v=="1":
        name=input("Enter name: ")
        surname = input("Enter surname: ")
        city = input("Enter city: ")
        country = input("Enter country: ")
        birthday = input("Enter birthday: ")
        with connection.cursor() as cursor:
            insert = f"INSERT INTO `toopeople` (name, surname,city,country,birthday) VALUES ('{name}', '{surname}', '{city}', '{country}', '{birthday}')"
            cursor.execute(insert)
            connection.commit()

    if v=="2":
        choose=input("What you want change: ")
        replace=input("And call what you want enter: ")
        idd=int(input("Enter id peron: "))
        with connection.cursor() as cursor:
            create_table = f"UPDATE `toopeople` SET {choose} = '{replace}' WHERE id = {idd}"
            cursor.execute(create_table)
            connection.commit()

    if v=="3":
        fir=input("Enter named colum: ")
        sec=input("Enter and what you want delete: ")
        with connection.cursor() as cursor:
            create_table = f"DELETE FROM `toopeople` WHERE {fir}='{sec}'"
            cursor.execute(create_table)
            connection.commit()

```

Задание 6

Добавьте к приложению возможность сохранять результаты работы фильтров в файл. Например, результат работы фильтра по отображению всех людей или результат работы фильтра по отображению людей из одного города.

```

file.close()

if v=="5":
    with connection.cursor() as cursor:
        select_all = "select name from people.toopeople"
        cursor.execute(select_all)
        result = cursor.fetchall()
        for row in result:
            print(row)
            file = open(os.path.join("based", "camber.json"), "a")
            json.dump(row, file)
            file.close()

```

