

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.21

**Взаимодействие с базами данных SQLite3 с помощью языка
программирования Python**

По дисциплине «Теории программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

Цель работы: приобретение навыков по работе с базами данных SQLite3 с помощью языка программирования Python версии 3.x.

Ход работы

1. Создал новый собственный репозиторий. Ссылка на репозиторий: https://github.com/Dmitry-15/2.21_laba.
2. С помощью команды `git clone` клонировал удаленный репозиторий на свой ПК. Дополнил файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
3. После ознакомления с теоретическим материалом приступил к выполнению заданий.

Пример 1

1. Для начала выполнил пример.

```
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> python primer.py add -n "Сидоров д. в." -р "Тренер" -у "2012"
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> python primer.py display
+-----+-----+-----+-----+
| No |      Ф.И.О.      |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Сидоров д. в.    |      Тренер      |      2012     |
+-----+-----+-----+-----+
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> 
```

Рисунок 1 – Результат выполнения примера

2. Приступил к выполнению индивидуальных заданий.

Индивидуальное задание 1

1. Условие задания: для своего варианта лабораторной работы 2.17 необходимо реализовать хранение данных в базе данных SQLite3. Информация в базе данных должна храниться не менее чем в двух таблицах.

```
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> python wasd.py add -z "стрелец" -n "сидоров ю.в." -у "14.12.2005"
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> python wasd.py display
+-----+-----+-----+-----+
| No |      ФИО      |      Знак зодиака      |      Дата рождения      |
+-----+-----+-----+-----+
|  1 | плотников д.в. |      лев      |      15.08.2002     |
+-----+-----+-----+-----+
|  2 | бобров н.в.    |      стрелец      |      19.12.2002     |
+-----+-----+-----+-----+
|  3 | иванов п.н.    |      стрелец      |      11.12.2003     |
+-----+-----+-----+-----+
|  4 | сидоров ю.в.   |      стрелец      |      14.12.2005     |
+-----+-----+-----+-----+
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> 
```

Рисунок 2 – Результат выполнения первого индивид. задания

2. С помощью СУБД DB Browser просмотрел в базе данных две таблицы.

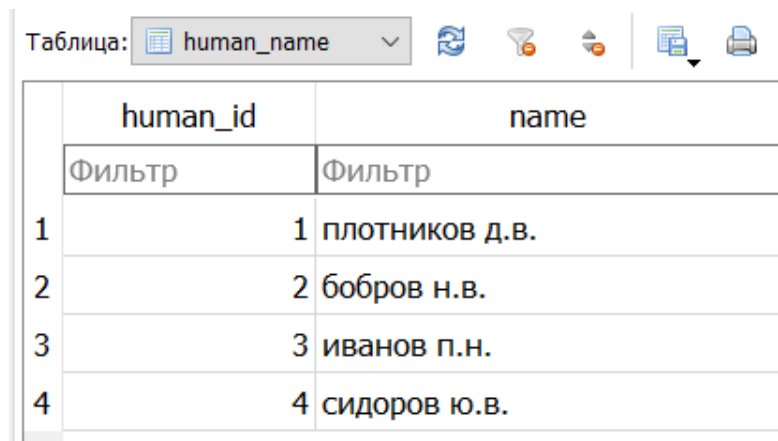


Таблица: human_name	
human_id	name
Фильтр	Фильтр
1	1 плотников д.в.
2	2 бобров н.в.
3	3 иванов п.н.
4	4 сидоров ю.в.

Рисунок 3 – Таблица только с ФИО людей

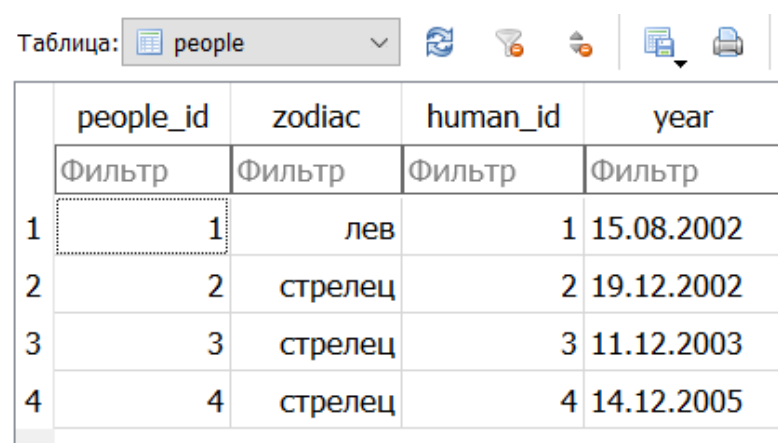


Таблица: people			
people_id	zodiac	human_id	year
Фильтр	Фильтр	Фильтр	Фильтр
1	1 лев	1	15.08.2002
2	2 стрелец	2	19.12.2002
3	3 стрелец	3	11.12.2003
4	4 стрелец	4	14.12.2005

Рисунок 4 – Таблица с полной информацией о людях

Индивидуальное задание 2

1. Условие задания: самостоятельно изучите работу с пакетом python-psycopg2 для работы с базами данных PostgreSQL. Для своего варианта лабораторной работы 2.17 необходимо реализовать возможность хранения данных в базе данных СУБД PostgreSQL. Информация в базе данных должна храниться не менее чем в двух таблицах.

2. Для начала установил PostgreSQL и настроил все параметры для работы, а также создал базу данных.

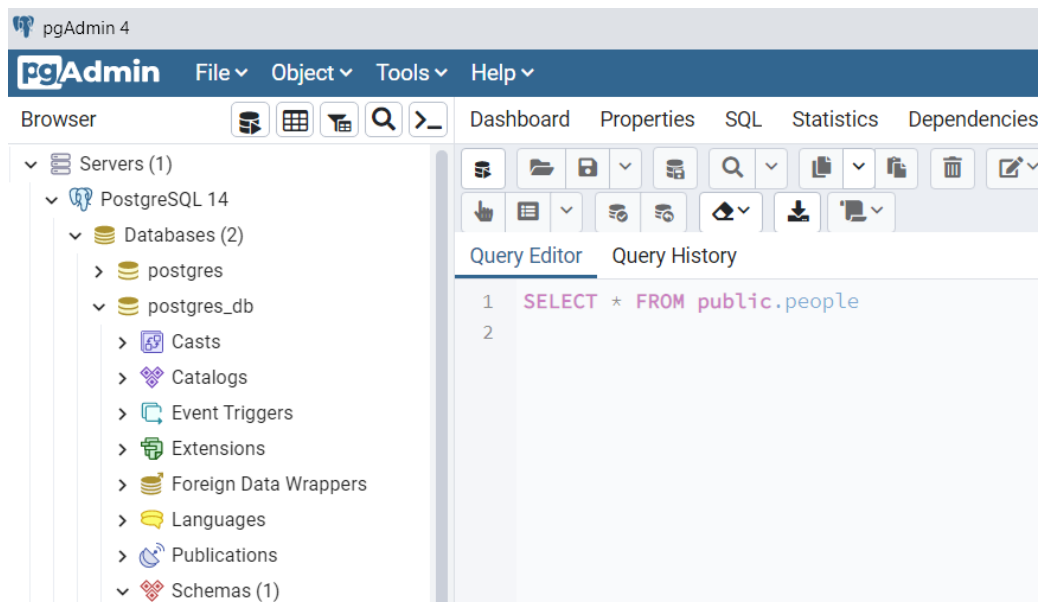


Рисунок 5 – Созданная база данных

3. Дополнительно установил пакет `psycopg2` и затем добавил людей в таблицу.

```
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> python main.py add -n "Плотников Д.В." -z "лев" -yr "15.08.2002"
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21> python individ2.py add -n "Якубович Т.С.." -z "стрелец" -yr "11.12.2002"
PS C:\Users\Plotnikov\PycharmProjects\laba_2.21>
```

Рисунок 6 – Добавление людей в таблицу

4. Проверил в созданной БД наличие таблицы.

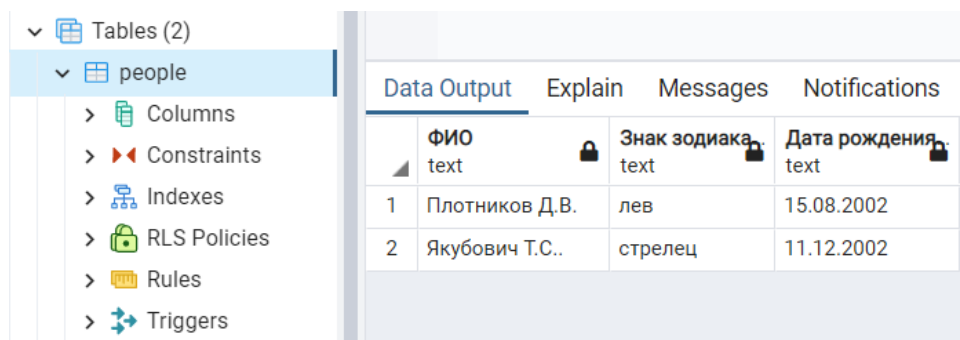


Рисунок 7 – Полученная таблица

Контрольные вопросы:

1. Каково назначение модуля `sqlite3`?

Модуль `sqlite3` предназначен для взаимодействия с СУБД SQLite.

2. Как выполняется соединение с базой данных SQLite3? Что такое курсор базы данных?

Объект соединения создается с помощью функции `connect()`.

Курсор SQLite3 – это метод объекта соединения. Для выполнения

инструкций SQLite3 сначала устанавливается соединение, а затем создается объект курсора с использованием объекта соединения.

3. Как подключиться к базе данных SQLite3, находящейся в оперативной памяти компьютера?

С помощью функции `:memory:`

4. Как корректно завершить работу с базой данных SQLite3?

Сначала импортируется модуль `sqlite3`, а затем определяется функция с именем `sql_connection`. Внутри функции у нас есть блок `try`, где функция `connect()` возвращает объект соединения после установления соединения. В случае возникновения ошибок при установке соединения с базой данных выполняются операторы блока `except`, в котором в данном случае просто печатается содержимое объекта ошибки. После этого вне зависимости от того возникло или нет исключение по работе с базой данных, выполняются операторы блока `finally`, в котором соединение закрывается. Закрывание соединения необязательно, но это хорошая практика программирования, поэтому вы освобождаете память от любых неиспользуемых ресурсов.

5. Как осуществляется вставка данных в таблицу базы данных SQLite3? Чтобы вставить данные в таблицу, используется оператор `INSERT INTO`.

6. Как осуществляется обновление данных таблицы базы данных SQLite3?

Чтобы обновить данные в таблице, просто создайте соединение, затем создайте объект курсора с помощью соединения и, наконец, используйте оператор `UPDATE` в методе `execute()`.

7. Как осуществляется выборка данных из базы данных SQLite3?

Оператор `SELECT` используется для выбора данных из определенной таблицы. Если вы хотите выбрать все столбцы данных из таблицы, вы можете использовать звездочку (*).

8. Каково назначение метода `rowcount`?

SQLite3 `rowcount` используется для возврата количества строк, которые были затронуты или выбраны последним выполненным SQL-запросом.

9. Как получить список всех таблиц базы данных SQLite3?

Чтобы перечислить все таблицы в базе данных SQLite3, вы должны запросить данные из таблицы `sqlite_master`, а затем использовать `fetchall()` для получения результатов из инструкции `SELECT`.

10. Как выполнить проверку существования таблицы как при ее добавлении, так и при ее удалении?

Чтобы проверить, не существует ли таблица уже, мы используем `IFNOT EXISTS` с оператором `CREATE TABLE`

11. Как выполнить массовую вставку данных в базу данных SQLite3?

Метод `executemany` можно использовать для вставки нескольких строк одновременно.

12. Как осуществляется работа с датой и временем при работе с базами данных SQLite3

В базе данных Python SQLite3 мы можем легко хранить дату или время, импортируя модуль `datetime`.

Вывод: в ходе выполнения лабораторной работы успешно приобретены навыки по взаимодействию с базами данных SQLite3 с помощью языка программирования Python 3.x.