

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

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.21

Тема: «Взаимодействие с базами данных SQLite3 с помощью
языка программирования Python»

Выполнил студент группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: исследовать возможности взаимодействия с базами данных SQLite3 с помощью языка программирования Python.


Ход работы:

1. Создать общедоступный репозиторий с лицензией MIT и языком Python.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 umarkochkarov ▾

Repository name *

/ lb2.21 ✓

Great repository names are short and memorable. Need inspiration? How about [upgraded-lamp?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  main as the default branch. Change the default name in your [settings](#).



You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Клонировать репозиторий на ПК:

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.21
$ git clone https://github.com/umarkochkarov/lb2.21.git
Cloning into 'lb2.21'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование репозитория

3. Организовать репозиторий в соответствии с моделью ветвления git-flow:

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.21/lb2.21 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/erken/Desktop/python/Ла62.21/lb2.21/.git/hooks]
```

Рисунок 3. Организация репозитория в соответствии с моделью ветвления git-flow

4. Проработка примеров лабораторной работы:

```
(env) C:\Users\erken\Desktop\python\Ла62.21\lb2.21\prim>python primer.py add --name="Иван Иванов" --post="Главный инженер" --year=2012

(env) C:\Users\erken\Desktop\python\Ла62.21\lb2.21\prim>python primer.py display
```

No	Ф.И.О.	Должность	Год
1	Иван Иванов	Главный инженер	2012

Рисунок 4. Выполнение примера

5. Индивидуальные задания:

1) 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.

```
C:\Users\erken\Desktop\python\Лаб2.21\lb2.21\ind>python ind.py display
```

№	Пункт назначения	Номер маршрута	Тип поезда
1	Moskow	656	Скоростной
2	Saint-P	111	Грузовой

```
C:\Users\erken\Desktop\python\Лаб2.21\lb2.21\ind>python ind.py select --type="Скоростной"
```

№	Пункт назначения	Номер маршрута	Тип поезда
1	Moskow	656	Скоростной

Рисунок 5. Индивидуальное задание

2) 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.

```
C:\Users\erken\Desktop\python\Лаб2.21\lb2.21\ind>python ind2.py add -d="Saint-P" -t="Грузовой" -n="111"
```

```
C:\Users\erken\Desktop\python\Лаб2.21\lb2.21\ind>python ind2.py display
```

№	Пункт назначения	Номер маршрута	Тип поезда
1	Moskow	656	Скоростной
2	Saint-P	111	Грузовой
3	Saint-P	111	Грузовой

```
C:\Users\erken\Desktop\python\Лаб2.21\lb2.21\ind>
```

Рисунок 6. Индивидуальное задание повышенной сложности

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. Заккрытие 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: были исследованы возможности взаимодействия с базами данных SQLite3 с помощью языка программирования Python.