

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 23(2.21)
Взаимодействие с базами данных SQLite3 с помощью языка
программирования Python
по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Злыгостев И.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2022

Цель работы: исследовать базовые возможности взаимодействия с базами данных SQLite3 с помощью языка программирования Python 3.x.

1. Изучив методические указания и разобрав примеры, приступил к выполнению индивидуальных заданий.
2. Реализовал две функции, отвечающие за создание соединения с базой данных и за создание таблицы в указанной базе данных.

```
def sql_connection(file):  
    try:  
        con = sqlite3.connect(file)  
        return con  
    except Error:  
        print(Error)  
  
def sql_table(con):  
    cursor_obj = con.cursor()  
    cursor_obj.execute(  
        """  
        CREATE TABLE IF NOT EXISTS flights (  
        "№" integer PRIMARY KEY autoincrement,  
        "Место прибытия" text,  
        "Номер самолёта" text,  
        "Тип" text)  
        """)  
    con.commit()
```

Рисунок 23.1 – Дополнение кода программы

```
(demo-2.21) C:\Users\zligo\Documents\GitHub\demo-2.21>python Задание1.py display mydatabase.db  
[(1, 'London', 'RF-86222', 'Airbus'), (2, 'Moscow', 'RF-86123', 'Airbus')]
```

Рисунок 23.2 – Результат выполнения программы

	№	Место прибытия	Номер самолёт	Тип
1	1	London	RF-86222	Airbus
2	2	Moscow	RF-86123	Airbus

Рисунок 23.3 – Проверка через графический интерфейс СУБД

3. Затем приступил к выполнению задания повышенного уровня сложности.
4. Для этого дополнительно изучил библиотеку psycopg2.

```
connection = psycopg2.connect(  
    user="postgres",  
    password="123asdqwezxcD",  
    host="127.0.0.1",  
    port="5432",  
    database="mydatebase")  
sql_table(connection)
```

Рисунок 23.4 – Дополнение кода

```
(demo-2.21) C:\Users\zligo\Documents\GitHub\demo-2.21>python Задание2.py add -s Berlin -v Boeing -n RF-863005
```

Рисунок 23.5 – Ввод данных через командную строку

	Data Output	Explain	Messages	Notifications
	Место прибытия text	Номер самолёта text	Тип text	
1	London	RF-8621	Airbus	
2	Moscow	RF-8312312	Airbus	
3	Berlin	RF-863005	Boing	

Рисунок 23.6 – Проверка правильности выполнения через графический интерфейс

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 `IF NOT 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 были приобретены навыки по взаимодействию с базами данных SQLite3 с помощью языка программирования Python 3.x.