

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №3
Взаимодействие с базами данных
SQLite3 с помощью языка программирования Python
По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Галяс Д. И. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2022

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

Ссылка на репозиторий: <https://github.com/DIMITRY-GALYAS1/Laba-2.21.git>

Ход работы:

1. Создал репозиторий и клонировал его на компьютер.
2. Выполнил пример.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import argparse
import sqlite3
import typing as t
from pathlib import Path

def display_workers(staff: t.List[t.Dict[str, t.Any]]) -> None:
    """
    Отобразить список работников.
    """
    # Проверить, что список работников не пуст.
    if staff:
        # Заголовок таблицы.
        line = '+--{}--{}--{}--{}--+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 8
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                "No",
                "Ф.И.О.",
                "Должность",
                "Год"
            )
        )
```

Рисунок 1. Пример

3. Далее выполнил первое индивидуальное задание.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
Для своего варианта лабораторной работы 2.17 необходимо реализовать хранение данных в
базе данных SQLite3.
"""

import sqlite3
import typing as t
from pathlib import Path
import argparse

def create_data(database_path: Path) -> None:
    """Создать базу данных"""
    conn = sqlite3.connect(database_path)
    cursor = conn.cursor()
    # Создать таблицу с ФИО студентов
    cursor.execute(
        """
        CREATE TABLE IF NOT EXISTS student_name (
            student_id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT NOT NULL
        )
        """
    )
    # Создать таблицу с полной информацией о студентах
    cursor.execute(
        """
        CREATE TABLE IF NOT EXISTS students (
            student_id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT NOT NULL,
            group_id INTEGER,
            progress INTEGER
        )
        """
    )
    # Добавить данные
    for row in data:
        add_student()
    if row is None:

```

Рисунок 2. Код первого задания

4. Проверил работу кода.

```

PS C:\Users\lizeq\PycharmProjects\pythonProject2.21> python individual_1.py display
+-----+-----+-----+-----+
| No |          ФИО          |      Группа      |      Успеваемость      |
+-----+-----+-----+-----+
| 1 | Игорь | 1 | 2 4 4 4 4 |
+-----+-----+-----+-----+
| 2 | Василий | 2 | 5 4 4 4 4 |
+-----+-----+-----+-----+

```

Рисунок 3. Отображение введенных данных

5. Затем проверил созданную таблицу данных.

	student_id	name	group	progress
	Фильтр	Фильтр	Фильтр	Фильтр
1	1	Игорь	1	2 4 4 4 4
2	2	Василий	2	5 4 4 4 4

Рисунок 4. Таблица записей

6. Выполнил второе индивидуальное задание.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
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.
"""

import argparse
import psycopg2
import typing as t
from pathlib import Path

def connect():
    conn = psycopg2.connect(
        user="postgres",
        password="12345",
        host="localhost",
        port="5432")

    return conn

def adding(
    name: str,
    group: int,
    progress: str
) -> None:
    """Добавить студента в базу данных"""
```

Рисунок 5. Второе индивидуальное задание

```
PS C:\Users\lizeq\PycharmProjects\pythonProject2.21> python individual_2.py add -n "Дима" -g "2" -p "5 5 5 5 5"
```

Рисунок 6. Проверка работы кода

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: в ходе выполнения лабораторной работы приобрел навыки по работе с базами данных SQLite3 с помощью языка программирования Python версии 3.x.