

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.21
дисциплины «Анализ данных»

Выполнил:
Говоров Егор Юрьевич
2 курс, группа ИВТ-Б-О-22-1,
27.04.2024 «Информатика и
вычислительная техника»,
направленность (профиль)
«Информатика и вычислительная
техника», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой_____ Дата защиты_____

Ставрополь, 2023 г.

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

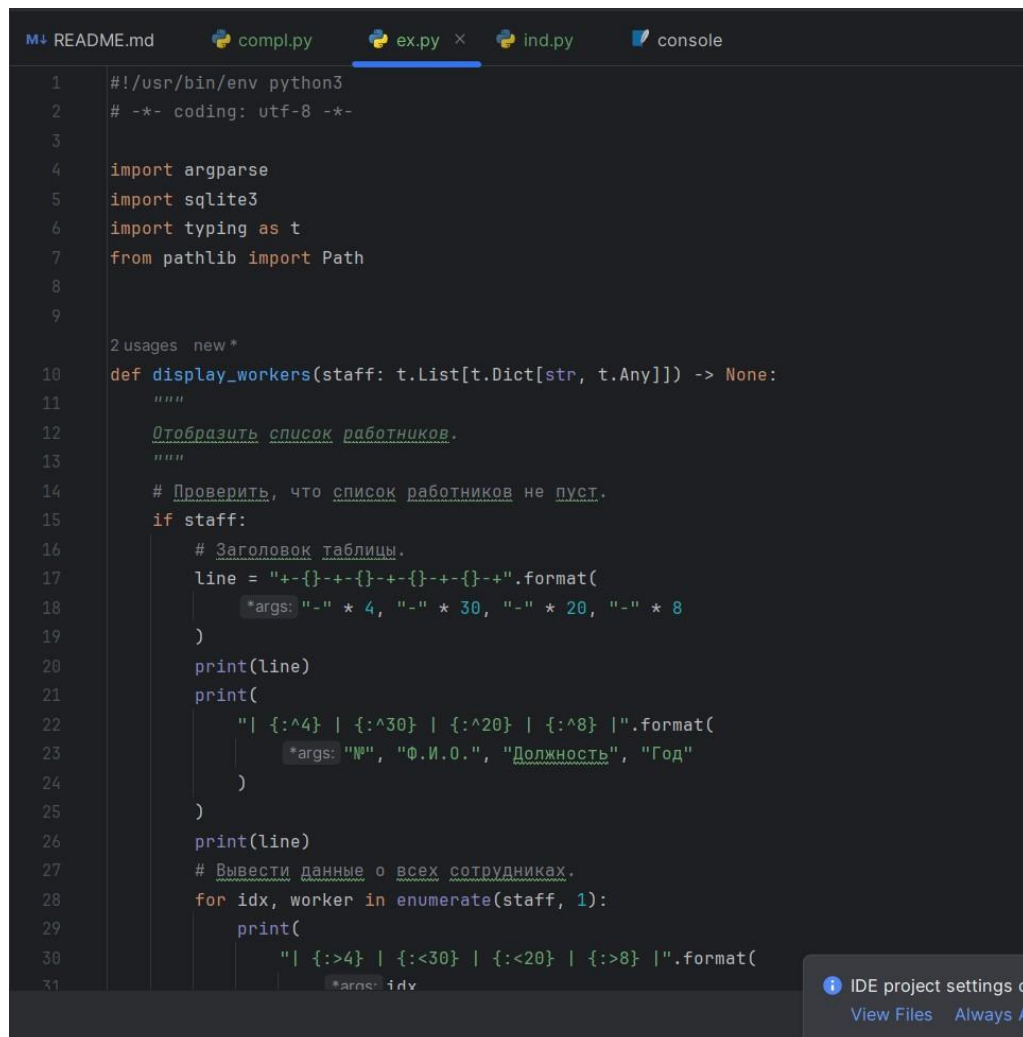
Цель: исследовать базовые возможности системы управления базами данных SQLite3 в языке Python.

Ход работы:

Задание 1. Создал проект PyCharm в папке репозитория. Приступил к работе с примером. Добавил новый файл primer.py.

Условие примера: Для примера лабораторной работы 2.17 реализуйте возможность хранения данных в базе данных SQLite3.

Решение: При решении задачи будем хранить данные о должностях в таблице posts, а данные о работниках в таблице workers. При каждом запуске программы будет проверяться наличие файла базы данных, и если такого файла нет, то он будет создан вместе с соответствующими таблицами. Для работы с датой и временем используются средства языка SQL SQLite3.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import argparse
5  import sqlite3
6  import typing as t
7  from pathlib import Path
8
9
10 2 usages new *
11 def display_workers(staff: t.List[t.Dict[str, t.Any]]) -> None:
12     """
13     Отобразить список работников.
14     """
15     # Проверить, что список работников не пуст.
16     if staff:
17         # Заголовок таблицы.
18         line = "+-{}-+-{}-+-{}-+-{}-+".format(
19             *args: "-" * 4, "-" * 30, "-" * 20, "-" * 8
20         )
21         print(line)
22         print(
23             "| {:^4} | {:^30} | {:^20} | {:^8} |".format(
24                 *args: "№", "Ф.И.О.", "Должность", "Год"
25             )
26         )
27         print(line)
28         # Вывести данные о всех сотрудниках.
29         for idx, worker in enumerate(staff, 1):
30             print(
31                 "| {:>4} | {:<30} | {:<20} | {:>8} |".format(
32                     *args: idx
```

Рисунок 1. Выполнение примера

```
Govorov-7 D:\Users\Admin\Рабочий стол\Data_Analyze 1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import argparse
5 import psycopg2
6 import typing as t
7
8 usage
9 def create_db() -> None:
10     """
11     Создать базу данных PostgreSQL.
12     """
13
14     conn = psycopg2.connect(
15         dbname="postgres",
16         user="postgres",
17         password="1234",
18         host="localhost",
19         port=5432,
20     )
21     cursor = conn.cursor()
22     cursor.execute('''
23         CREATE TABLE IF NOT EXISTS flights (
24             id SERIAL PRIMARY KEY,
25             destination TEXT NOT NULL,
26             flight_number TEXT NOT NULL,
27             aircraft_type TEXT NOT NULL
28         )
29     ''')
30     conn.commit()
31     conn.close()
```

Рисунок 2. Код усложненного задания

```
Командная строка
C:\Users\slime>cd C:\универ2\git\Data_Analyze_Govorov-7\ind
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py add
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py -a
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py -h
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py h
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py --add-flight
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py -a
Введите название пункта назначения: Москва-Питер
Введите номер рейса: 1333
Введите тип самолета: Грузовой
C:\универ2\git\Data_Analyze_Govorov-7\ind>python hard.py -a
Введите название пункта назначения: Новгород-Ставрополь
Введите номер рейса: 546475
Введите тип самолета: Пассажирский
C:\универ2\git\Data_Analyze_Govorov-7\ind>
```

Рисунок 3. Добавление данных

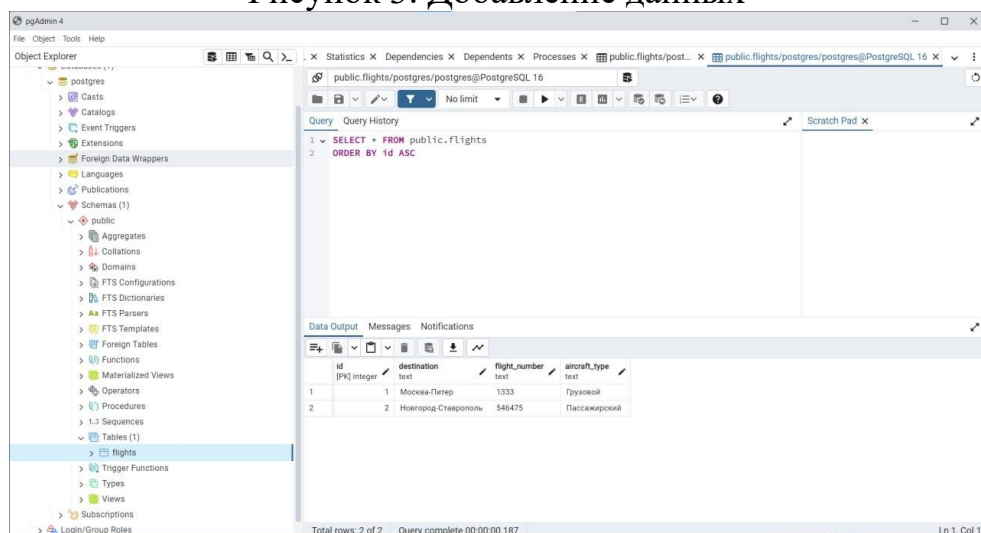


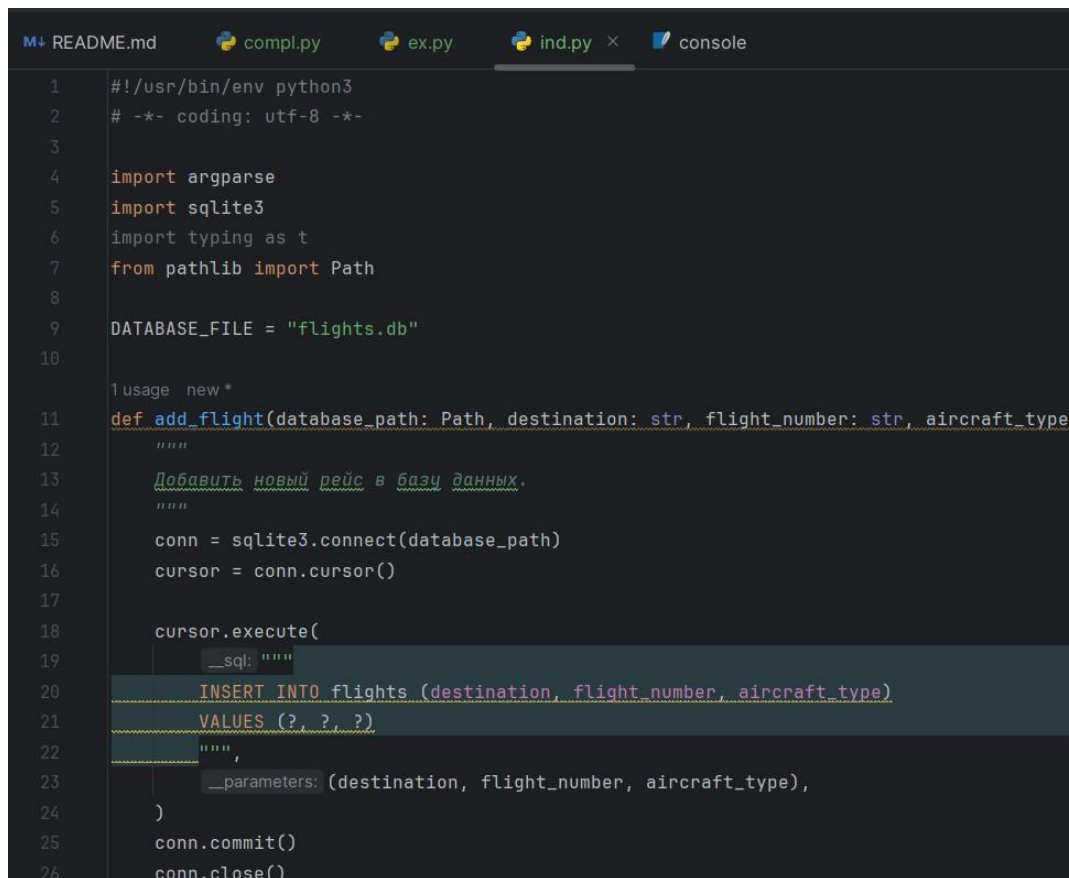
Рисунок 4. Добавленные в pgAdmin данные

Индивидуальное задание

Вариант 5

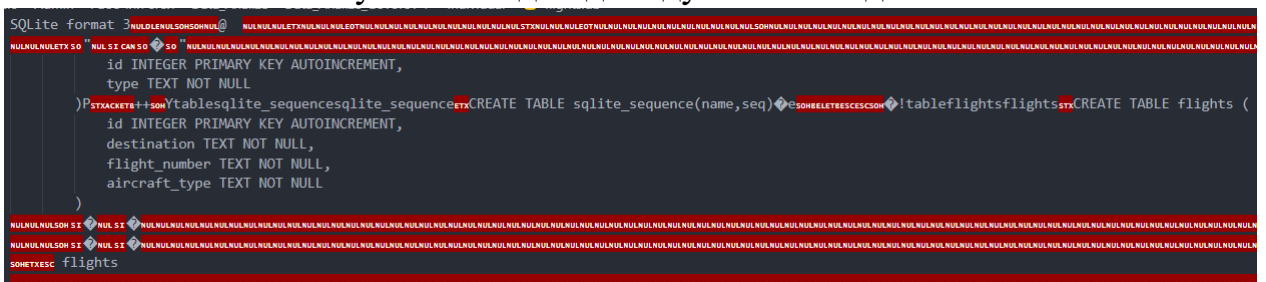
Создал новый файл под названием ind.py.

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.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import argparse
5  import sqlite3
6  import typing as t
7  from pathlib import Path
8
9  DATABASE_FILE = "flights.db"
10
11  usage new *
12
13  def add_flight(database_path: Path, destination: str, flight_number: str, aircraft_type: str) -> None:
14      """
15      Добавить новый рейс в базу данных.
16      """
17      conn = sqlite3.connect(database_path)
18      cursor = conn.cursor()
19
20      cursor.execute(
21          """
22          INSERT INTO flights (destination, flight_number, aircraft_type)
23          VALUES (?, ?, ?)
24          """
25      ,
26      _parameters: (destination, flight_number, aircraft_type),
27      )
28      conn.commit()
29      conn.close()
```

Рисунок 3. Код индивидуального задания



```
SQLite format 3
id INTEGER PRIMARY KEY AUTOINCREMENT,
type TEXT NOT NULL
)
CREATE TABLE sqlite_sequence(name,seq)
CREATE TABLE flights (
id INTEGER PRIMARY KEY AUTOINCREMENT,
destination TEXT NOT NULL,
flight_number TEXT NOT NULL,
aircraft_type TEXT NOT NULL
)
flights
```

Рисунок 4. Результат программы

Ответы на контрольные вопросы:

1. Каково назначение модуля sqlite3?

Модуль sqlite3 предоставляет интерфейс для взаимодействия с базами данных SQLite из программ, написанных на языке Python. Он позволяет создавать, управлять и выполнять запросы к базам данных SQLite.

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

Соединение с базой данных SQLite3 выполняется с использованием функции sqlite3.connect(). Эта функция возвращает объект соединения. Курсор базы данных (cursor) используется для выполнения SQL-запросов и получения результатов.

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

Чтобы подключиться к базе данных SQLite3, находящейся в оперативной памяти, нужно использовать специальное имя файла базы данных ":memory:" при вызове connect.

4. Как корректно завершить работу с базой данных SQLite3?

Чтобы корректно завершить работу с базой данных SQLite3, вызовите метод close объекта соединения.

5. Как осуществляется вставка данных в таблицу базы данных SQLite3?

Вставка данных в таблицу выполняется с использованием SQL-запроса INSERT INTO.

6. Как осуществляется обновление данных таблицы базы данных SQLite3?

Обновление данных в таблице выполняется с использованием SQL-запроса UPDATE.

7. Как осуществляется выборка данных из базы данных SQLite3?

Выборка данных выполняется с использованием SQL-запроса SELECT.

8. Каково назначение метода rowcount?

Метод `rowcount` возвращает количество строк, затронутых последним выполненным SQL-запросом.

9. Как получить список всех таблиц базы данных SQLite3?

Используйте запрос к таблице `sqlite_master`.

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

Вы можете использовать условие `IF NOT EXISTS` при создании таблицы и запрос к `sqlite_master` при проверке существования.

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

Для массовой вставки данных обычно используется метод `executemany`.

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

SQLite3 поддерживает тип данных `DATE` и `TIMESTAMP` для хранения даты и времени. При вставке и выборке данных с использованием этих типов следует использовать соответствующий формат.

Вывод: исследовал базовые возможности системы управления базами данных SQLite3 в языке Python.