

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:
Магдаев Даламбек Магомедович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

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

Цель: изучить взаимодействие с базами данных SQLite3 с помощью языка программирования Python.

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Проработал пример лабораторной работы: Для примера 1 лабораторной работы 2.17 реализуйте возможность хранения данных в базе данных SQLite3.

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py -h

display          Display all workers
select           Select the workers

options:
-h, --help       show this help message and exit
--version        show program's version number and exit
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py display -h

options:
-h, --help       show this help message and exit
--db DB          The database file name
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py display --db prim.db
Список работников пуст.
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py add --db prim.db -n Magdaev -p student -y 2022
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py add --db prim.db -n Motovilov -p student -y 2020
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py display --db prim.db
+-----+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+-----+
|  1 | Magdaev                  | student              | 2022          |
+-----+-----+-----+-----+-----+
|  2 | Motovilov                | student              | 2020          |
+-----+-----+-----+-----+-----+
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python prim.py select --db prim.db -P 3
+-----+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+-----+
|  1 | Motovilov                | student              | 2020          |
+-----+-----+-----+-----+-----+
```

Рисунок 1. Работа программы prim1.py

3. Выполнил индивидуальное задание: 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.

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind.py -h
usage: students [-h] [--version] {add,display,select} ...

positional arguments:
  {add,display,select}
    add                Add a new student
    display            Display all students
    select            Select the students

options:
  -h, --help            show this help message and exit
  --version            The main parser

(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind.py display -h
usage: students display [-h] [--db DB]

options:
  -h, --help            show this help message and exit
  --db DB              The database file name

(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind.py display --db ind.db
Список студентов пуст.
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind.py add --db ind.db --name "Magdaev D.M." --group 5 --grade "3 4 5 4 5"
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind.py add --db ind.db --name "Ivanov I.I." --group 7 --grade "3 3 4 3 4"
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind.py display --db ind.db
```

№	Ф.И.О.	Группа	Успеваемость
1	Magdaev D.M.	5	3 4 5 4 5
2	Ivanov I.I.	7	3 3 4 3 4

Рисунок 2. Результат работы программы ind.py

4. Выполнил задание повышенной сложности: 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.

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind_hard.py -h
usage: students [-h] [--version] {add,display,select} ...

positional arguments:
  {add,display,select}
    add                Add a new student
    display            Display all students
    select            Select the students

options:
  -h, --help            show this help message and exit
  --version            The main parser

(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind_hard.py display -h
usage: students display [-h] [--db DB]

options:
  -h, --help            show this help message and exit
  --db DB              The database file name

(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind_hard.py display --db ind_hard.db
```

№	Ф.И.О.	Группа	Успеваемость
1	Ivanov I.I.	13	3 5 4 3 4
2	Magdaev D.M.	11	4 5 4 3 4
3	Motovilov V.B.	15	5 5 4 4 3

```
(pythonProject) PS C:\Users\dalam\Desktop\projects\study\data_analysis\lab_2.21> python ind_hard.py select --select=13
```

№	Ф.И.О.	Группа	Успеваемость
1	Ivanov I.I.	13	3 5 4 3 4

Рисунок 3. Результат работы программы hard.py

Ответы на контрольные вопросы:

1. Каково назначение модуля sqlite3?

Инструменты для работы с конкретной СУБД не являются базовыми командами и объектами самого языка. Обычно они подключаются через импорт модуля или библиотеки. Непосредственно модуль sqlite3 – это API к СУБД SQLite. Своего рода адаптер, который переводит команды, написанные на Питоне, в команды, которые понимает SQLite. Как и наоборот, доставляет ответы от SQLite в python-программу.

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

Объект соединения создается с помощью функции connect() с параметром – путем к базе данных. Для взаимодействия с базой данных SQLite3 в Python необходимо создать объект cursor. Создать его можно с помощью метода cursor() .

Курсор SQLite3 – это метод объекта соединения. Для выполнения инструкций SQLite3 сначала устанавливается соединение, а затем создается объект курсора с использованием объекта соединения. Теперь можно использовать объект cursor для вызова метода execute() для выполнения любых SQL-запросов.

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

```
con = sqlite3.connect(':memory:')
```

4. Как корректно завершить работу с базой данных SQLite3?

```
conn.close()
```

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

```
# Сам запрос
```

```
sql_query = "INSERT INTO my_table (column1, column2, column3)
```

VALUES (?, ?, ?)"

Данные для вставки

data_to_insert = ('value1', 'value2', 'value3') # Выполнение SQL-запроса с данными cursor.execute(sql_query, data_to_insert)

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 .

sqlite_master – это главная таблица в SQLite3, которая хранит все таблицы.

```
conn = sqlite3.connect('example.db') cursor = conn.cursor()
```

```
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
```

```
# Получение списка таблиц tables = cursor.fetchall()
```

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 результате выполнения работы были исследованы взаимодействия с базами данных SQLite3 с помощью языка программирования python.