

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 16

**Взаимодействие с базами данных SQLite3 с помощью языка
программирования Python**

По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Бобров Н. В. « » _____ 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.

Ход работы:

1. Создал общедоступный репозиторий и клонировал его на локальный сервер.
2. Ознакомившись с теоретическим материалом, выполнил примеры, создав для них отдельный модуль.
3. Приступил к выполнению индивидуального задания.

```
9  def selecting(con, student):
10      cur= con.cursor()
11      cur.execute("""SELECT * FROM students WHERE "Успеваемость" = ?""", (student,))
12      print(cur.fetchall())
13
14
15  def table(con):
16      cur= con.cursor()
17      cur.execute("SELECT * FROM students")
18      print(cur.fetchall())
19
20
21  def adding(con, name, group, grade):
22      cur= con.cursor()
23      cur.execute(f"""INSERT INTO students("ФИО", "Группа", "Успеваемость")
24      VALUES(?, ?, ?);""", (name, group, grade))
25      con.commit()
26
```

Рисунок 1 – Выборка данных

4. Затем объявил две функции, отвечающие за подключение к базе данных и создание в ней таблицы.

```

28 def sql_connection(file):
29     try:
30         con = sqlite3.connect(file)
31         return con
32     except Error:
33         print(Error)
34
35
36 def sql_table(con):
37     cursor_obj = con.cursor()
38     cursor_obj.execute(
39         """
40         CREATE TABLE IF NOT EXISTS students (
41             "№" integer PRIMARY KEY autoincrement,
42             "ФИО" text,
43             "Группа" text,
44             "Успеваемость" text)
45         """
46     )
47     con.commit()

```

Рисунок 2 – Подключение к БД и создание в ней таблицы

5. Проверил результат выполнения кода.

```

PS C:\TPA\Laba3> python individual1.py add -n "Бобров Николай" -g "30" -gr "5 5 5 5 5"
PS C:\TPA\Laba3> python individual1.py display
+-----+-----+-----+-----+
| No |          ФИО          |      Группа      | Успеваемость |
+-----+-----+-----+-----+
|  1 | Бобров Николай      | 30               | 5 5 5 5 5 |
+-----+-----+-----+-----+

```

Рисунок 3 – Ввод данных

6. Затем, с помощью СУБД DB Browser проверил таблицу графически.

Таблица:

student_name

student_id	name
Фильтр	Фильтр
1 1	Бобров Нико...

Таблица:

students

student_id	group	grade
Фильтр	Фильтр	Фильтр
1 1	30	5 5 5 5 5

Рисунок 4 – Отображение записей в СУБД

7. Приступил к выполнению второго индивидуального задания.

8. Установил PostgreSQL, затем внес необходимые параметры для работы с БД. Для этого в файлах с расширением .py прописал настройки для создания базы данных.

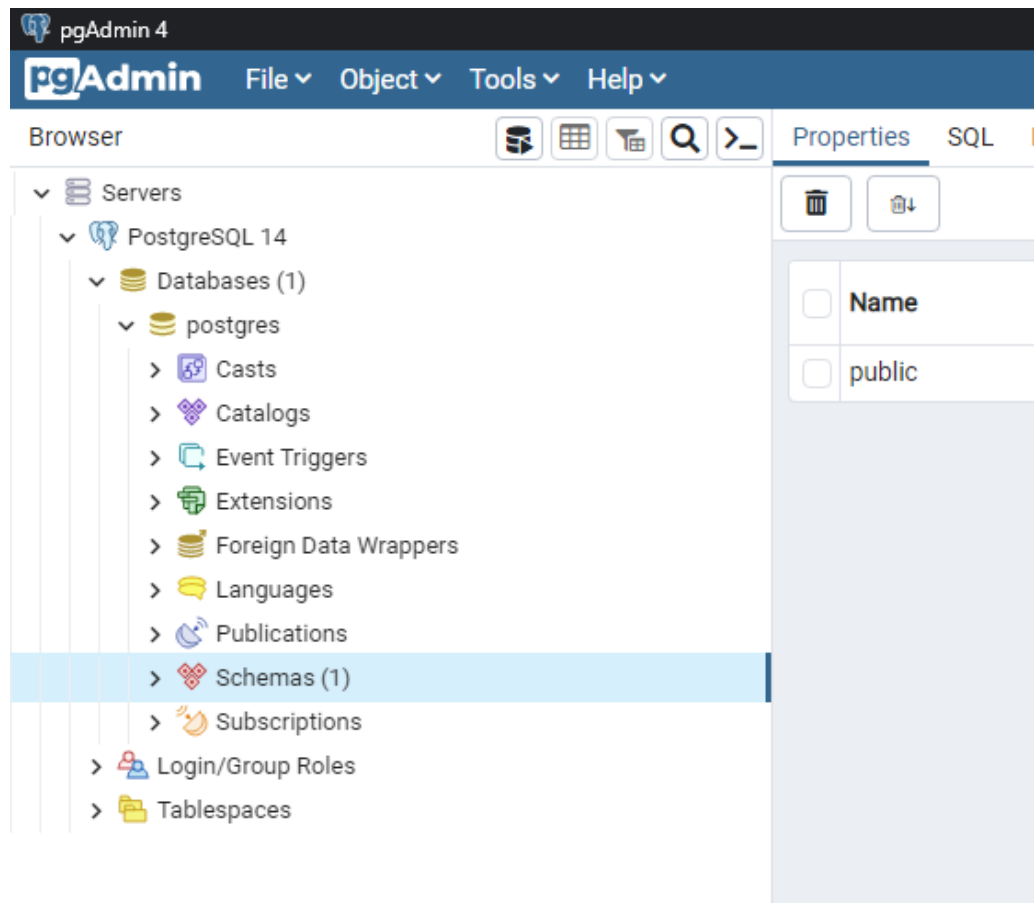


Рисунок 6 – Созданная база данных

9. Для выполнения индивидуального задания, установил библиотеку `rsusorg2`, затем в функции написал код для её работы.

```

26 def create_db() -> None:
27     cursor = connect().cursor()
28     # Создать таблицу с ФИО студентов
29     cursor.execute(
30         """
31         CREATE TABLE IF NOT EXISTS student_name (
32             student_id serial,
33             PRIMARY KEY(student_id),
34             name TEXT NOT NULL
35         )
36         """
37     )
38     # Создать таблицу с полной информацией о студентах
39     cursor.execute(
40         """
41         CREATE TABLE IF NOT EXISTS students (
42             student_id serial,
43             PRIMARY KEY(student_id),
44             group_id INTEGER NOT NULL,
45             grade TEXT NOT NULL,
46             FOREIGN KEY(student_id) REFERENCES student_name(student_id)
47             ON DELETE CASCADE ON UPDATE CASCADE
48         )
49         """
50     )

```

```

15 def connect():
16     connection = psycopg2.connect(
17         user="postgres",
18         password="b0brov5572",
19         host="127.0.0.1",
20         port="5432"
21     )
22
23     return connection

```

Рисунок 7 – Дополнение кода

10. Проверил работоспособность кода.

```

PS C:\TPA\Laba3> python task2.py add -n "Евгений Коршунов" -g "4" -gr "5 4 4 5 5"

```

Рисунок 8 – Вывод записи

Data Output	Explain	Messages	Notifications
<div> <div>ФИО</div> <div>text</div> </div>	<div> <div>Группа</div> <div>text</div> </div>	<div> <div>Успеваемость</div> <div>text</div> </div>	
1 Евгений Коршунов	4	5 4 4 5 5	

Рисунок 9 – Запись в таблице СУБД

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 были приобретены навыки по взаимодействию с базами данных SQLite3 с помощью языка программирования Python 3.x.