

Северо-Кавказский федеральный университет  
Институт математики и информационных технологий

**ОТЧЕТ**  
**о выполнении лабораторной работы №8**  
**по дисциплине**  
**«Основы Программной Инженерии»**

Выполнил:

**Ботвинкин Никита Сергеевич**

---

студент 2 курса, ПИЖ-б-о-21-1 группы  
бакалавриата «Программная инженерия»  
очной формы обучения

---

Ставрополь, 2023

## СКРИНШОТЫ ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО

```
PS C:\git\2_8lab> python Individualnoe.py add --db="my.db" --name="никита ботвинкин" --group="пиж" --marks=33455
PS C:\git\2_8lab> python Individualnoe.py show --db="my.db"
```

№	Ф.И.О.	Группа	Успеваемость
1	никита ботвинкин	пиж	3 3 4 5 5

Рисунок 8.1 – Добавление и просмотр записей

```
PS C:\git\2_8lab> python Individualnoe.py add --db="my.db" --name="никита неботвинкин" --group="пиж" --marks=33452
PS C:\git\2_8lab> python Individualnoe.py add --db="my.db" --name="неникита ботвинкин" --group="пиж" --marks=34554
PS C:\git\2_8lab> python Individualnoe.py show --db="my.db"
```

№	Ф.И.О.	Группа	Успеваемость
1	никита ботвинкин	пиж	3 3 4 5 5
2	никита неботвинкин	пиж	3 3 4 5 2
3	неникита ботвинкин	пиж	3 4 5 5 4

```
PS C:\git\2_8lab> python Individualnoe.py show_marks --db="my.db"
```

№	Ф.И.О.	Группа	Успеваемость
1	никита неботвинкин		3 3 4 5 2

Рисунок 8.2 – Просмотр студентов с оценкой 2

```
PS C:\git\2_8lab> python Individualnoe.py add --db="my.db" --name="ошибочная ошибка" --group="err" --marks=3554_
ошибка в количестве или значении оценок
PS C:\git\2_8lab> python Individualnoe.py add --db="my.db" --name="ошибочная ошибка" --group="err" --marks=355442_
ошибка в количестве или значении оценок
PS C:\git\2_8lab> python Individualnoe.py add --db="my.db" --name="ошибочная ошибка" --group="err" --marks=35547_
ошибка в количестве или значении оценок
PS C:\git\2_8lab> python Individualnoe.py show --db="my.db"
```

№	Ф.И.О.	Группа	Успеваемость
1	никита ботвинкин	пиж	3 3 4 5 5
2	никита неботвинкин	пиж	3 3 4 5 2
3	неникита ботвинкин	пиж	3 4 5 5 4

Рисунок 8.3 – Проверка на ошибку

```
PS C:\git\2_8lab> python postgre.py add --db="my" --name="никита" --group="пиж" --marks=34555
PS C:\git\2_8lab> python postgre.py add --db="my" --name="игорь" --group="пиж" --marks=34555
PS C:\git\2_8lab> python postgre.py show --db="my"
```

№	Ф.И.О.	Группа	Успеваемость
1	никита	пиж	3 4 5 5 5
2	игорь	пиж	3 4 5 5 5

```
PS C:\git\2_8lab> python postgre.py add --db="notmy" --name="неигорь" --group="пиж" --marks=34555
PS C:\git\2_8lab> python postgre.py add --db="notmy" --name="неникита" --group="пиж" --marks=34555
PS C:\git\2_8lab> python postgre.py show --db="notmy"
```

№	Ф.И.О.	Группа	Успеваемость
1	неигорь	пиж	3 4 5 5 5
2	неникита	пиж	3 4 5 5 5

Рисунок 8.4 – Добавление и просмотр записей в разные базы данных

```

PS C:\git\2_8lab> python postgre.py add --db="new" --name="хорошист" --group="пиж" --marks=44444
PS C:\git\2_8lab> python postgre.py add --db="new" --name="отличник" --group="пиж" --marks=55555
PS C:\git\2_8lab> python postgre.py show --db="new"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | хорошист | пиж | 4 4 4 4 4 |
| 2 | отличник | пиж | 5 5 5 5 5 |
+-----+-----+-----+-----+
PS C:\git\2_8lab> python postgre.py add --db="new" --name="ошибочный" --group="пиж" --marks=232324
ошибка в количестве или значении оценок
PS C:\git\2_8lab> python postgre.py add --db="new" --name="двоечник" --group="пиж" --marks=23224
PS C:\git\2_8lab> python postgre.py show_marks --db="new"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | двоечник | | 2 3 2 2 4 |
+-----+-----+-----+-----+
PS C:\git\2_8lab> python postgre.py show --db="new"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | хорошист | пиж | 4 4 4 4 4 |
| 2 | отличник | пиж | 5 5 5 5 5 |
| 3 | двоечник | пиж | 2 3 2 2 4 |
+-----+-----+-----+-----+

```

Рисунок 8.5 – Ошибочное значение и просмотр студента с оценкой «2»

```

PS C:\git\2_8lab> python postgre.py add --db="new" --name="иностранец" --group="непиж" --marks=24554
PS C:\git\2_8lab> python postgre.py show --db="new"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | хорошист | пиж | 4 4 4 4 4 |
| 2 | отличник | пиж | 5 5 5 5 5 |
| 3 | двоечник | пиж | 2 3 2 2 4 |
| 4 | иностранец | непиж | 2 4 5 5 4 |
+-----+-----+-----+-----+

```

Рисунок 8.6 – Добавление студента из другой группы

```

PS C:\git\2_8lab> python postgre.py show_marks --db="new"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | двоечник | | 2 3 2 2 4 |
| 2 | иностранец | | 2 4 5 5 4 |
+-----+-----+-----+-----+

```

Рисунок 8.7 – Просмотр студентов с двойками

## ОТВЕТЫ НА КОТРОЛЬНЫЕ ВОПРОСЫ

### 1. Каково назначение модуля sqlite3?

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?

Чтобы использовать SQLite3 в Python, прежде всего, вам нужно будет импортировать модуль sqlite3, а затем создать объект соединения, который соединит нас с базой данных и позволит нам выполнять операторы SQL. 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?

При создании соединения с SQLite3 автоматически создается файл базы данных, если он еще не существует. Этот файл базы данных создается на диске, мы также можем создать базу данных в оперативной памяти с помощью функции:

memory:with the connect.

Такая база данных называется базой данных в памяти.

### 4. Как корректно завершить работу с базой данных SQLite3?

Нужно применить изменения (db.commit()) и разорвать соединение (db.close())

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.

8. Каково назначение метода rowcount?

В 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`.