

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-
СКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.20

Тема: «Основы
работы с SQLite3»

Выполнил студент группы

ИВТ-б-о-21-1

Богадунов В.И. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Цель работы: исследовать базовые возможности системы управления базами данных SQLite3.

Ход работы:

1. Создал репозиторий в GitHub, дополнил правила в .gitignore для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

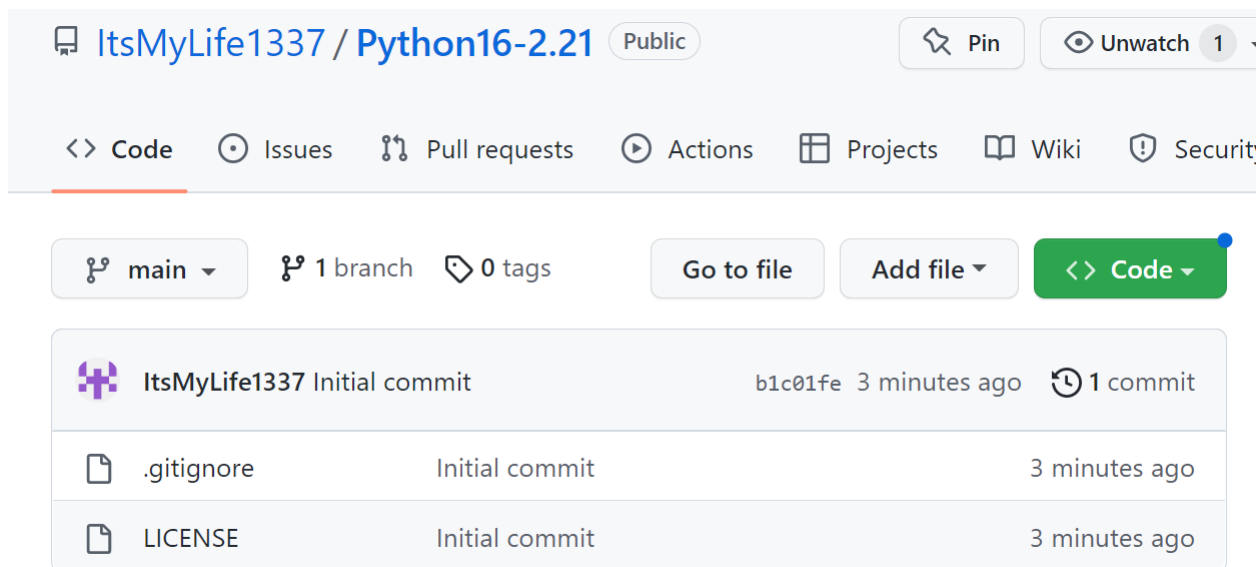


Рисунок 1.1 – Созданный репозиторий

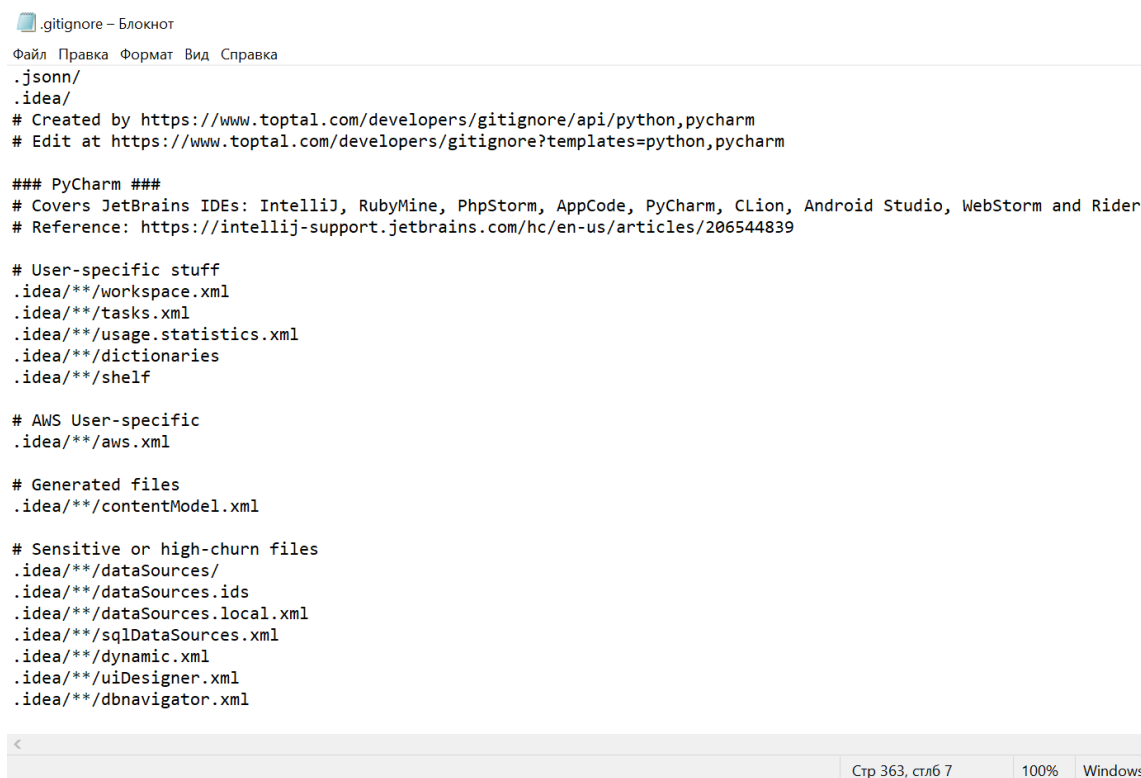


Рисунок 1.2 – Дополнил правила в .gitignore

```

c:\Users\Admin\Desktop\git\Python16-2.21>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Admin/Desktop/git/Python16-2.21/.git/hooks]

c:\Users\Admin\Desktop\git\Python16-2.21>git status

```

Рисунок 1.3 – Организация репозитория в соответствии с моделью ветвления git-flow

2. Проработал примеры ЛР.

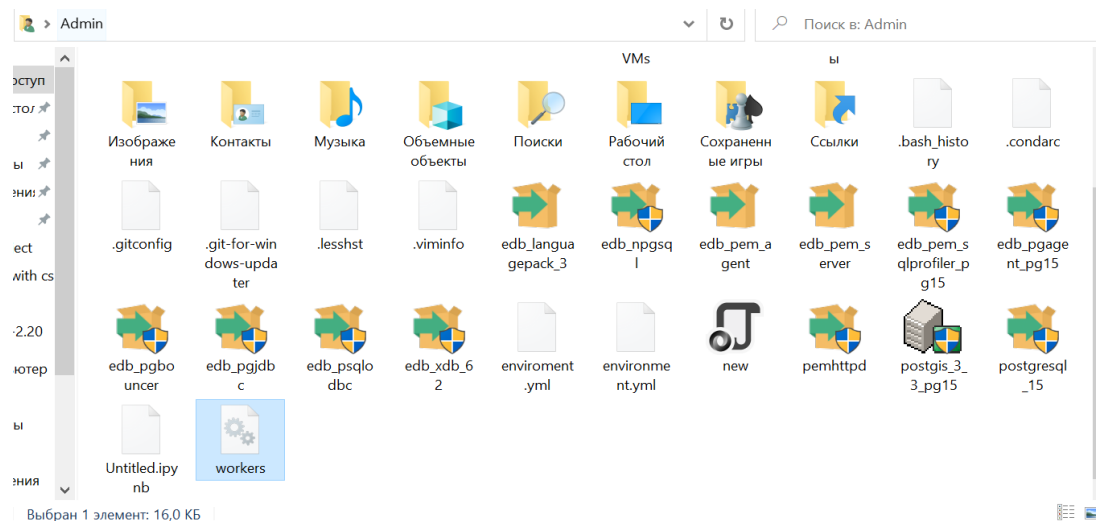


Рисунок 2 – Работа с примером

```

c:\Users\Admin\Desktop>python ind7.py add --db bd.db -n "Sis" -p "Tre" -y "2002"

c:\Users\Admin\Desktop>python ind7.py display --db workers.db
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Sis | Tre | 2222 |
+-----+-----+-----+-----+

c:\Users\Admin\Desktop>python ind7.py display --db db.db
Список работников пуст.

c:\Users\Admin\Desktop>python ind7.py display --db bd.db
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Sis | Tre | 2222 |
| 2 | Sis | Tre | 2222 |
| 3 | Sis | Tre | 2222 |
| 4 | Sis | Tre | 2222 |
| 5 | Sis | Tre | 2002 |
+-----+-----+-----+-----+

```

Рисунок 3 – Созданная база данных

таблица: workers Filter in...

worker_id	worker_name	post_id	worker_year
Фильтр	Фильтр	Фильтр	Фильтр
1	Борисенко БИ	1	2012

Рисунок 4 – Проверка отображения графически в приложении DB Browser

Индивидуальное задание №1. 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.

```
C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>python ind_1.py add --name="Сидоров Сидор" --group="12" --grade="4"

C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>python ind_1.py display
```

№	Ф.И.О.	Группа	Успеваемость
1	Сидоров Сидор	12	4

```
C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>
```

Рисунок 6 – Проверка работы индивидуального задания

students 06.04.2023 19:38 Файл "DB" 16 КБ

Рисунок 7 – Созданная база данных

DB Browser for SQLite - C:\Users\Admin\Desktop\git\Python16-2.21\Individuals\students.db

Файл Редактирование Вид Инструменты Справка

Новая база данных Открыть базу данных Записать изменения Отменить измен

Структура БД Данные Прагмы SQL

Таблица: students Filter in

student_id	student_name	group_id	student_grade
Фильтр	Фильтр	Фильтр	Фильтр
1	Сидоров Сидор	1	4

Рисунок 8 – Проверка записи данных в бд

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.

```
C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>python ind_2.py select --select="11"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Пётр Васильевич | 11 | 5 |
+-----+-----+-----+-----+

C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>python ind_2.py display
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Сидоров Сидор | 12 | 4 |
| 2 | Пётр Васильевич | 11 | 5 |
+-----+-----+-----+-----+

C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>python ind_2.py select --select="11"
+-----+-----+-----+-----+
| № | Ф.И.О. | Группа | Успеваемость |
+-----+-----+-----+-----+
| 1 | Пётр Васильевич | 11 | 5 |
+-----+-----+-----+-----+

C:\Users\Admin\Desktop\git\Python16-2.21\Individuals>
```

Рисунок 9 – Проверка работоспособности программы

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 обязательно, 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 результате лабораторной работы были исследованы базовые возможности взаимодействия с базами данных SQLite3 с помощью языка программирования Python.