МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

**АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО**

ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

**Отчет по лабораторной работе № 2.21**

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

**по дисциплине «Основы программной инженерии»**

Выполнила:

Образцова Мария Дмитриевна,

2 курс, группа ПИЖ-б-о-21-1, Проверил:

Доцент кафедры инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2023 г.

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

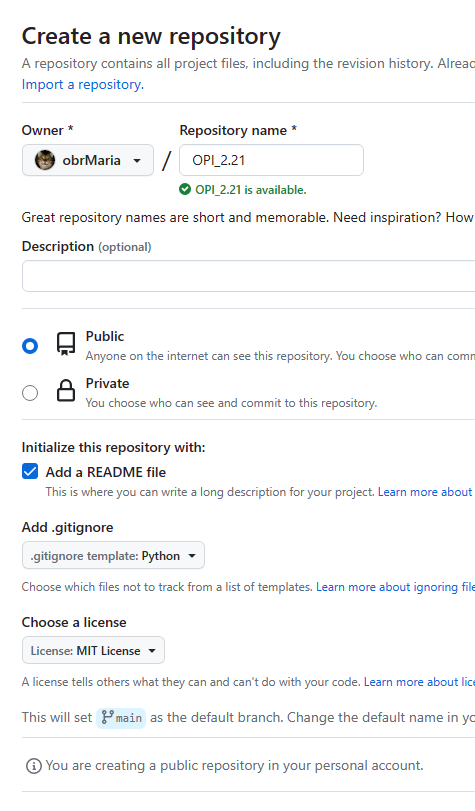


Рисунок 1 – создание репозитория

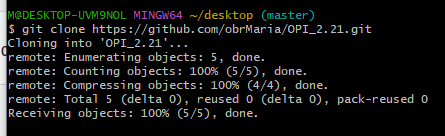


Рисунок 2 – клонирование репозитория

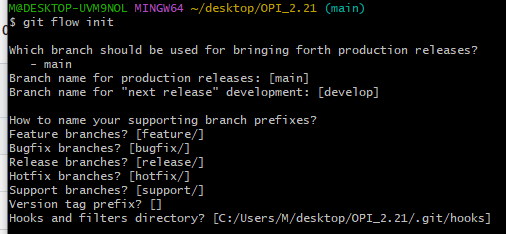


Рисунок 3 – создание ветки develop

Проработайте примеры лабораторной работы. Создайте для них отдельные модули языка Python. Зафиксируйте изменения в репозитории.



Рисунок 4 – Пример вывода классов

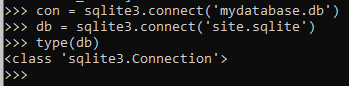


Рисунок 5 – Пример вызова функции connect()



Рисунок 6 – Пример использования курсора



Рисунок 7 – Пример создания базы данных

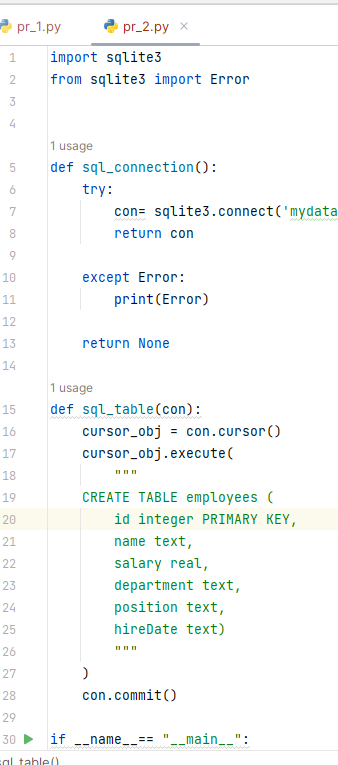
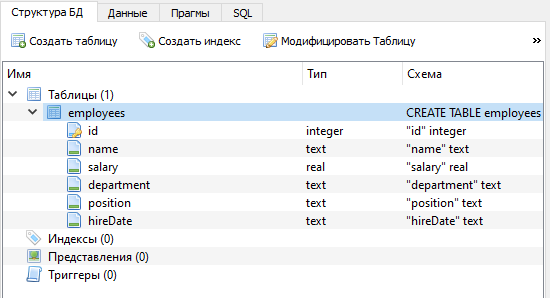


Рисунок 8 – Пример создания таблиц



Результат 9 – создания базы данных

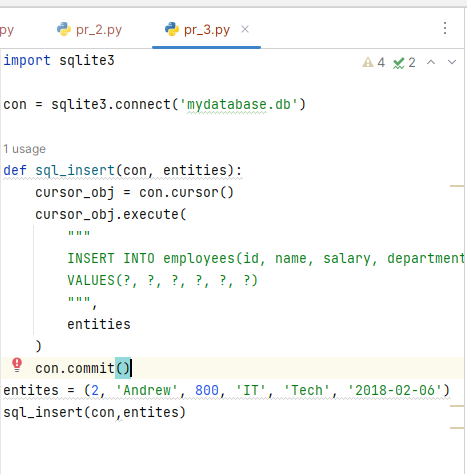


Рисунок 10 – Пример вставки данных в таблицу

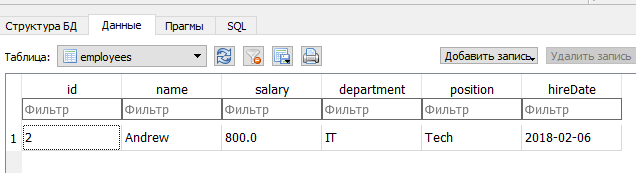


Рисунок 11– результат выполнения программы

.

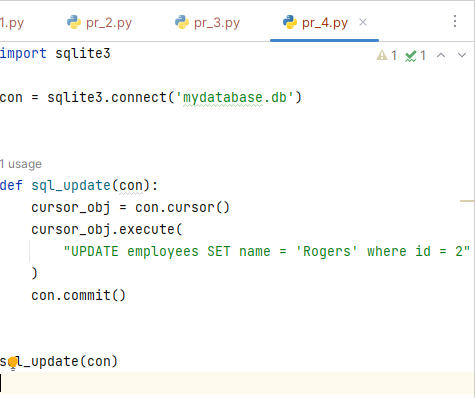


Рисунок 12 – Пример обновления данных в таблицах

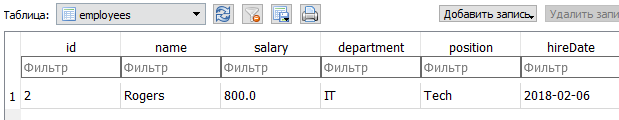


Рисунок 13– результат выполнения программы

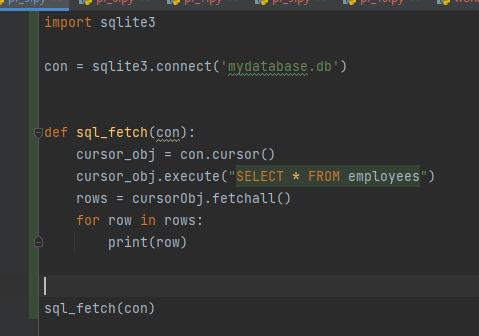


Рисунок 14 – Пример выборки данных из таблицы

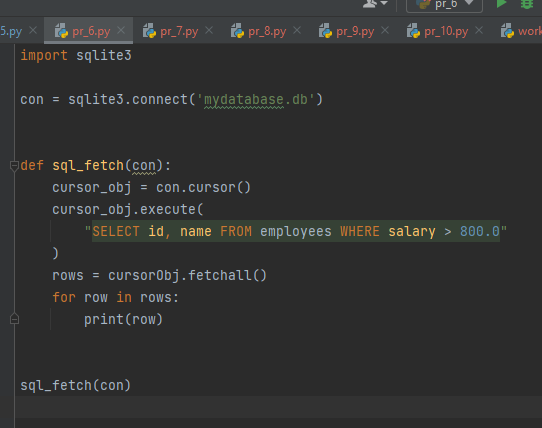


Рисунок 15 – Пример получения списка таблиц

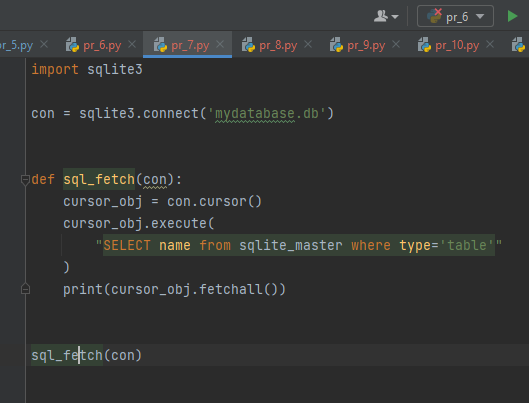


Рисунок 16 – Пример проверки существования таблицы

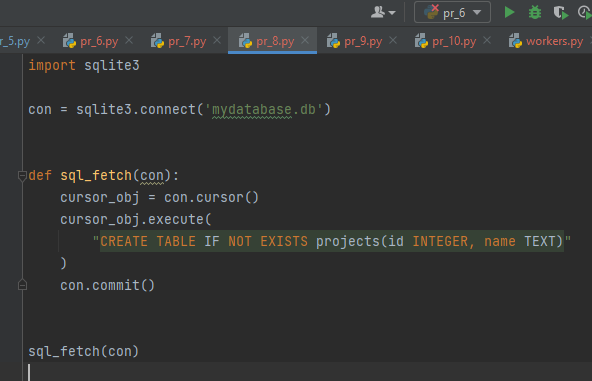


Рисунок 17 – Пример проверки существования таблицы

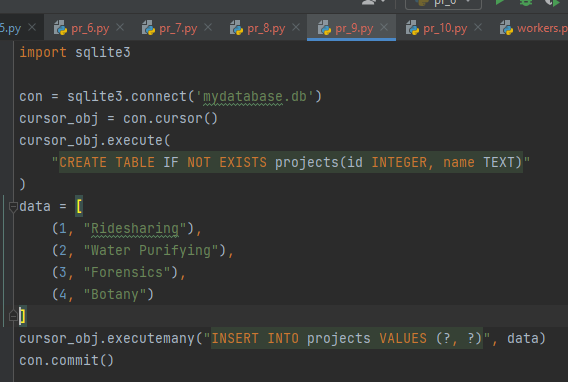


Рисунок 18 – Пример массовой вставки

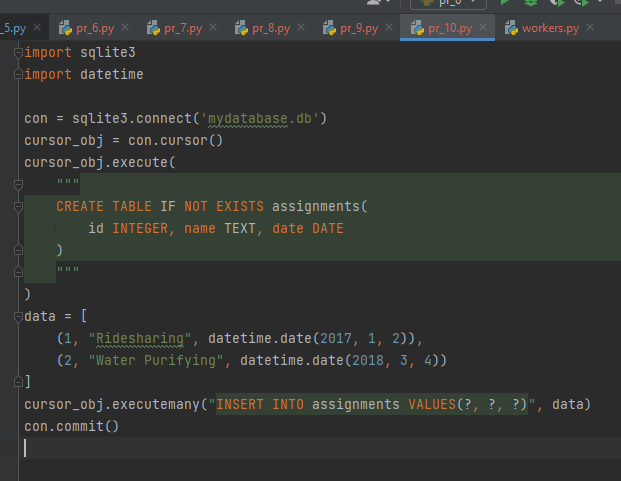


Рисунок 19 – Пример использования datetime



Рисунок 20 – Пример

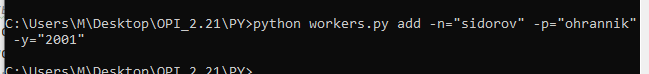


Рисунок 21 – запуск программы

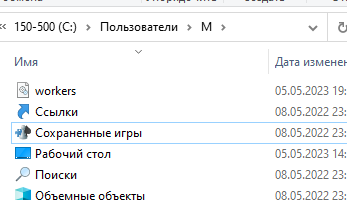


Рисунок 22 – результат выполнения примера

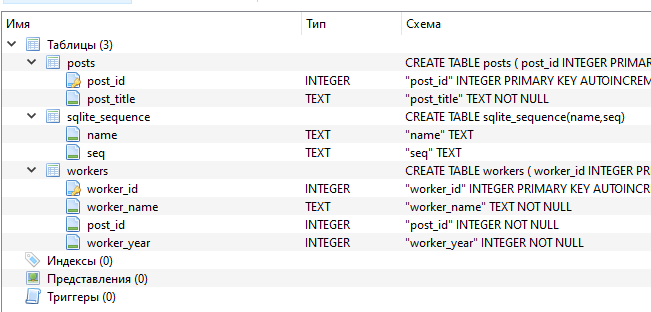


Рисунок 23 – полученная база данных

**Индивидуальные 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.

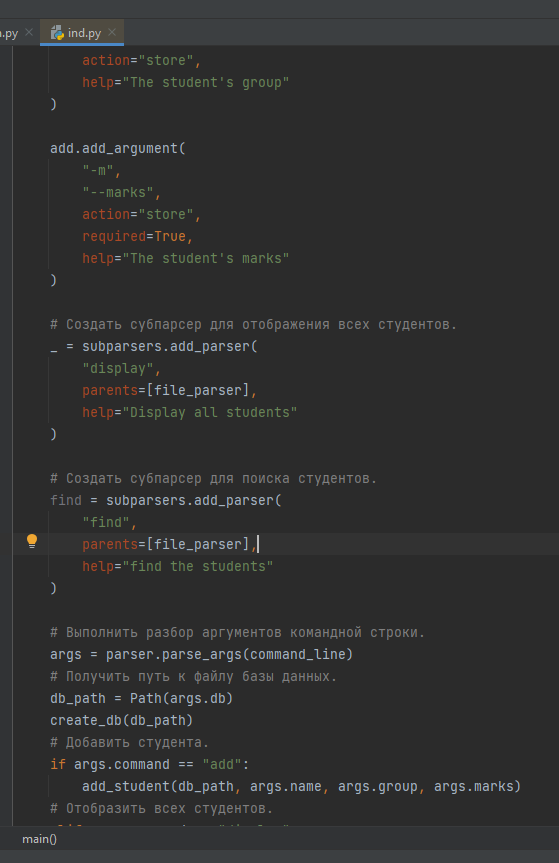


Рисунок 24 – индивидуальное задание



Рисунок 25 – запуск программы

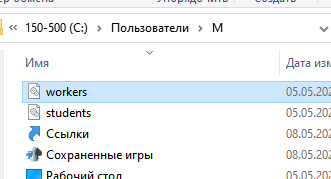


Рисунок 26 – результат выполнения

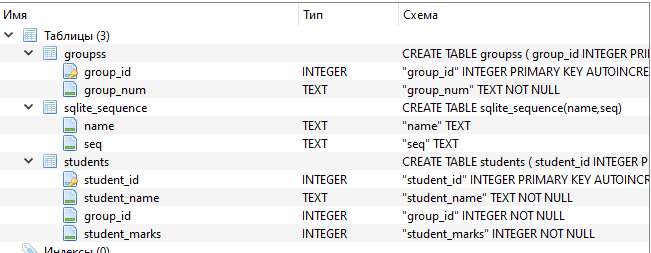


Рисунок 27 – база данных

**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.

Вопросы для защиты работы

# 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.

# 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 следующим образом:

# 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.

# Как корректно завершить работу с базой данных SQLite3?

После этого вне зависимости от того возникло или нет исключение по работе с базой данных, выполняются операторы блока finally, в котором соединение закрывается. Закрытие соединения необязательно, но это хорошая практика программирования, поэтому вы освобождаете память от любых неиспользуемых ресурсов.

# Как осуществляется вставка данных в таблицу базы данных SQLite3?

Чтобы вставить данные в таблицу, используется оператор INSERT INTO.

# 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 ().

# Как осуществляется выборка данных из базы данных SQLite3?

Оператор SELECT используется для выбора данных из определенной таблицы.

# Каково назначение метода rowcount?

SQLite3 rowcount используется для возврата количества строк, которые были затронуты или выбраны последним выполненным SQL-запросом.

# 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.

# Как выполнить проверку существования таблицы как при ее добавлении, так и при её удалении?

Чтобы проверить, не существует ли таблица уже, мы используем IF NOT EXISTS с оператором CREATE TABLE.

# Как выполнить массовую вставку данных в базу данных SQLite3?

Метод executemany можно использовать для вставки нескольких строк одновременно.

# 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.