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 №2.20**

# дисциплины «Основы кроссплатформенного программирования»

|  |
| --- |
| Выполнил:  Пинаева Диана Олеговна  2 курс, группа ИТС-б-о-22-1,  11.03.02 «Инфокоммуникационные технологии и системы связи», направленность (профиль)  «Инфокоммуникационные системы и сети», очная форма обучения  (подпись) |
| Руководитель практики: Воронкин Р. А., доцент кафедры инфокоммуникаций  (подпись) |

Отчет защищен с оценкой \_ Дата защиты

Ставрополь, 2023 г.

**Тема:** Основы работы с SQLite3

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

# Ход работы:

Задание 1.Создала общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами. Клонировала свой репозиторий на свой компьютер. Организовала свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

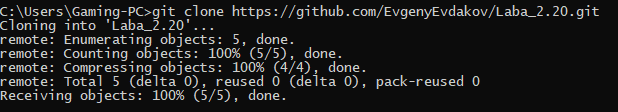


Рисунок 1. Клонирование репозитория

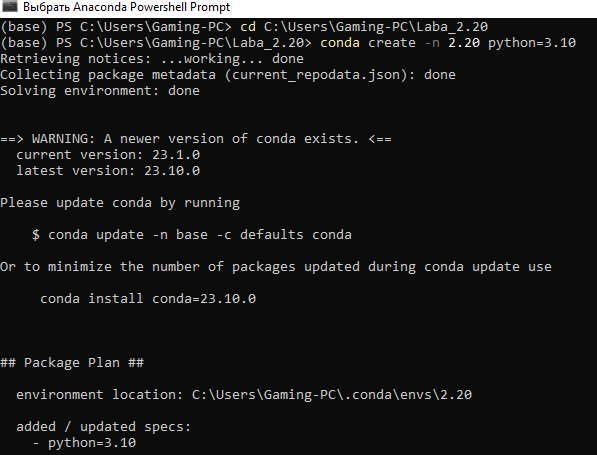
Задание 2.Создала виртуальное окружение conda и активировала его, также установила необходимые пакеты isort, black, flake8, pyinputplus.

Рисунок 2. Создание виртуального окружения

Задание 3.Решить задачу: выполнить в песочнице команды:



Рисунок 3. Команды для выполнения

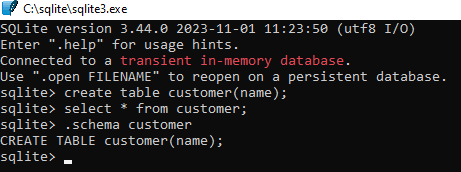


Рисунок 4. Выполнение первого общего задания Вот что здесь происходит:

1. Первая команда ( create table ) создает таблицу customer с единственным столбцом name .
2. Вторая команда ( select ) показывает содержимое таблицы customer (она пустая).
3. Третья команда ( .schema ) показывает список и структуру всех таблиц в базе.

create и select — это SQL-запросы, часть стандарта SQL. Запрос может занимать несколько строк, а в конце всегда ставится точка с запятой.

.schema — это специальная команда SQLite, не часть стандарта SQL.

Задание 4.Решить задачу: c помощью команды .help найти в песочнице команду, которая отвечает за вывод времени выполнения запроса.

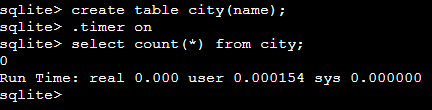


Рисунок 5. Выполнение второго общего задания

Задание 5.Решить задачу: загрузить файл city.csv в песочнице, затем выполнить запрос, который вернет число, получить число после выполнения запроса.

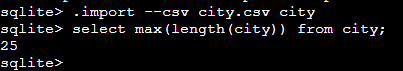


Рисунок 6. Выполнение третьего общего задания

Задание 6.Решить задачу: загрузить файл city.csv в песочницу с помощью команды .import , но без использования опции --csv . Эта опция появилась только в недавней версии SQLite (3.32, май 2020), так что полезно знать способ, подходящий для старых версий.

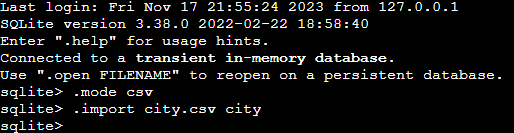


Рисунок 7. Выполнение четвертого общего задания

Задание 7.

Решить задачу: написать в песочнице запрос, который посчитает количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах. Вывести столбцы timezone и city\_count, отсортируйте по значению часового пояса:

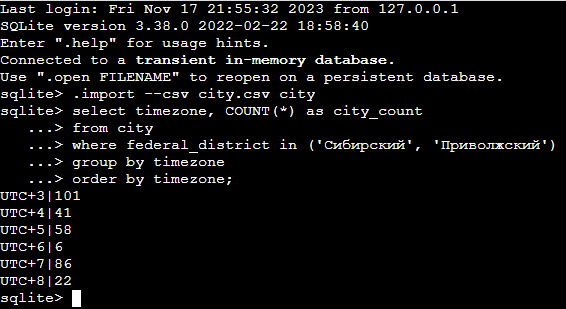


Рисунок 8. Выполнение пятого общего задания

Задание 8.Решить задачу: написать в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару. Указать в ответе названия этих трех городов через запятую в порядке удаления от Самары.

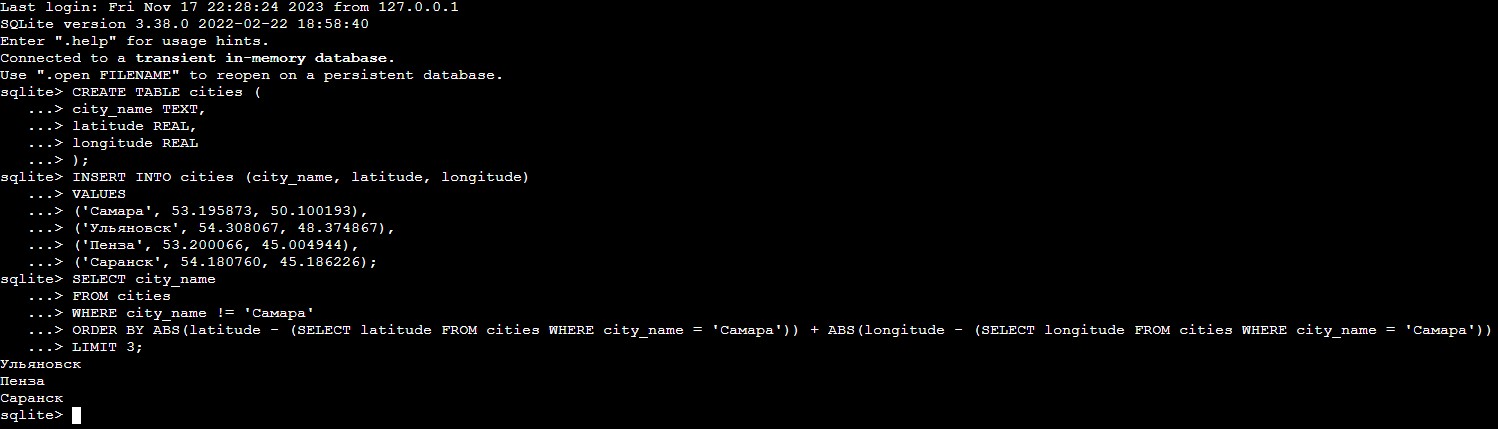


Рисунок 9. Выполнение шестого общего задания

Задание 9.Решить задачу: написать в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортировать по количеству городов по убыванию.

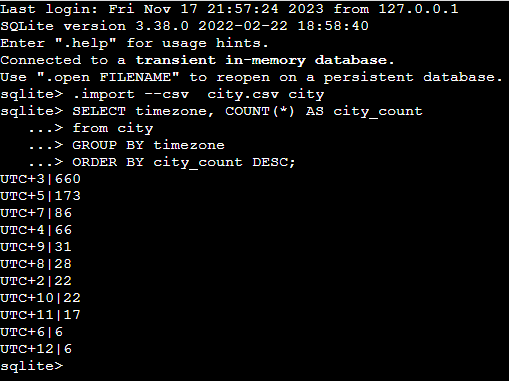


Рисунок 10. Выполнение седьмого общего задания Выполнить этот же запрос, но так, чтобы результат был

* в формате CSV
* с заголовками

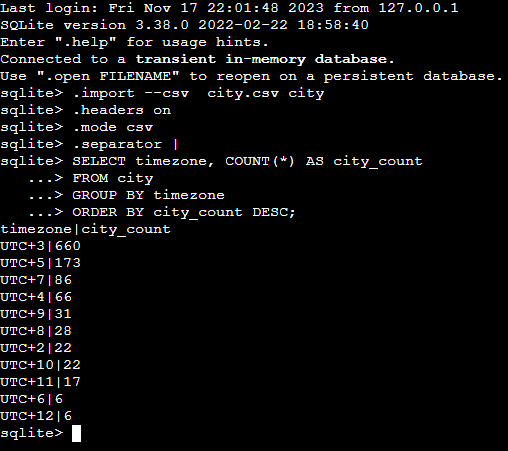


Рисунок 11. Выполнение седьмого общего с изменениями

**Индивидуальное задание**

**Вариант 10**

Условие задания:Загрузить в SQLite выбранный датасет в формате CSV (датасет можно найти на сайте Kaggle). Сформировать более пяти запросов к таблицам БД. Выгрузить результат выполнения запросов в форматы CSV и JSON.

Скачала датасет с сайта Kaggle под названием Billionaires Statistics Dataset, который содержит информацию о миллиардерах.

Создадим запросы к таблицам БЗ.

Для добавления файла json используем команду:



1. Выполнение запроса: Подсчитать количество миллиардеров в каждой стране:

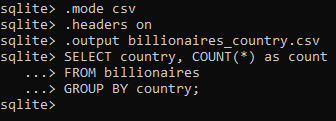


Рисунок 12. Выполнение первого запроса

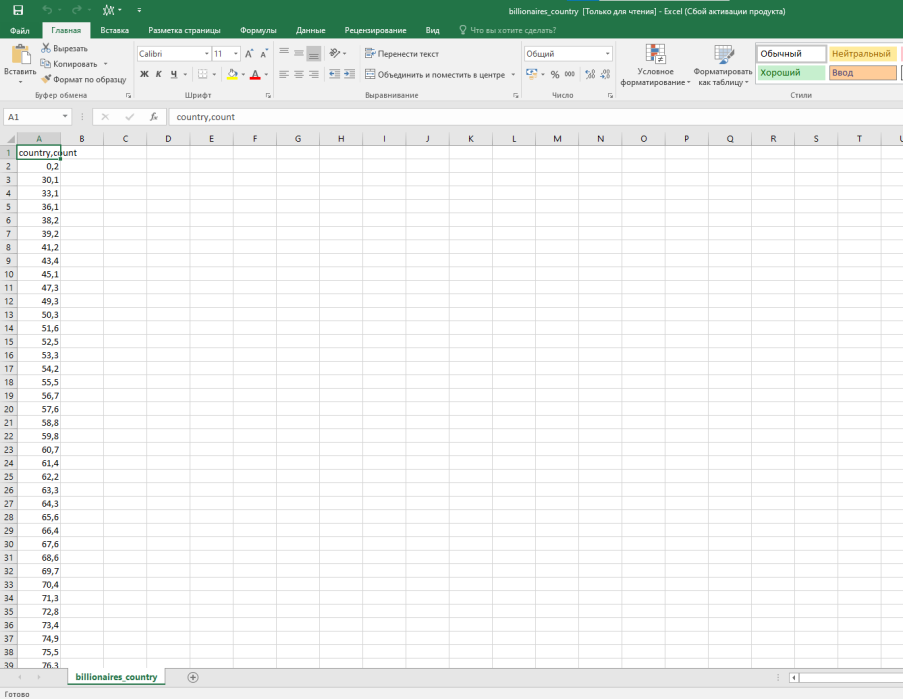


Рисунок 13. Файл CSV для первого запроса

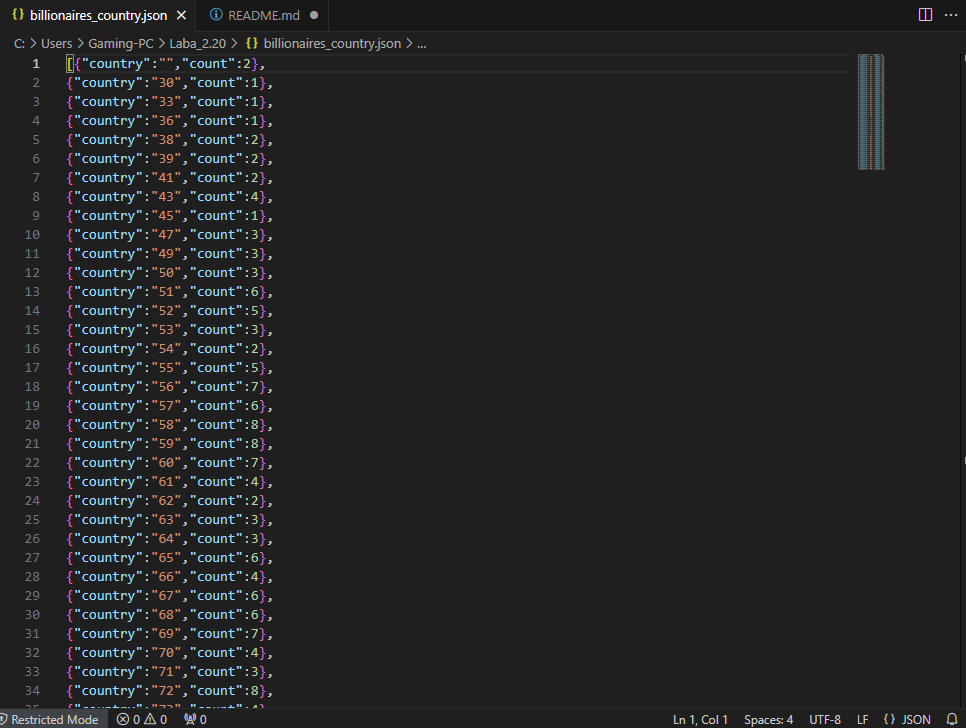


Рисунок 14. Файл JSON для первого запроса

1. Выполнение запроса: Вывести имена и состояния (net worth) миллиардеров с состоянием более 50 миллиардов:

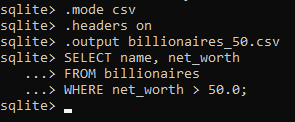


Рисунок 15. Выполнение второго запроса

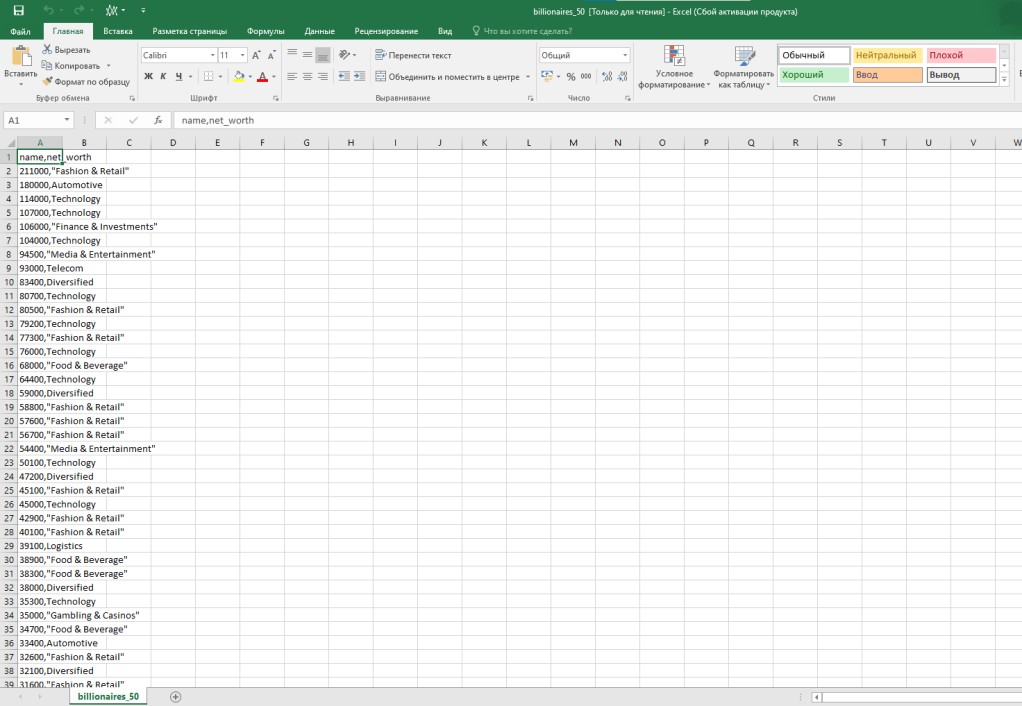


Рисунок 16. Файл CSV для второго запроса

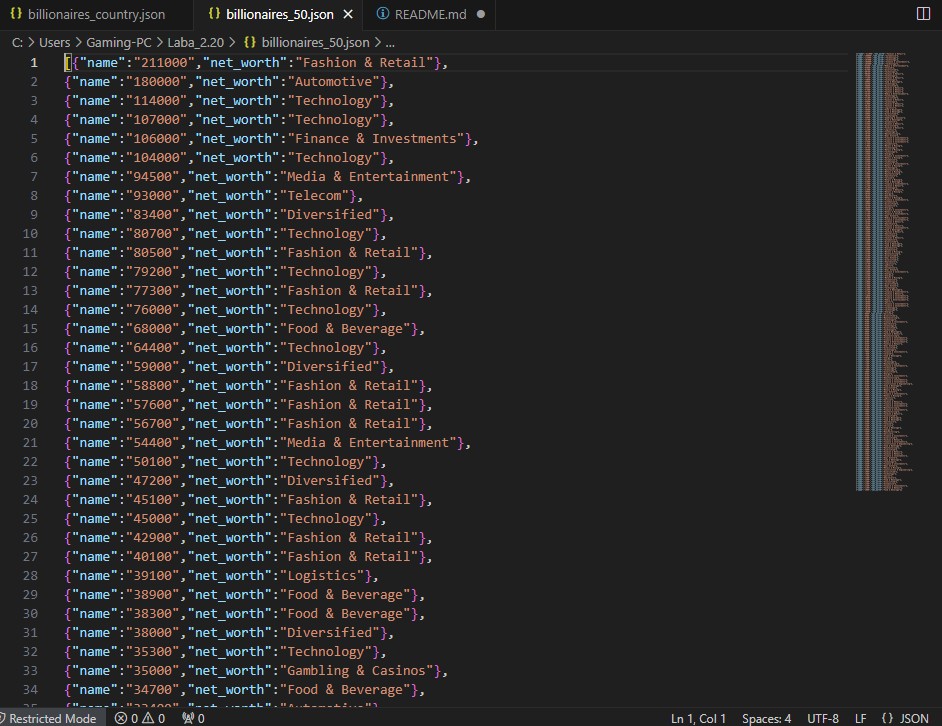


Рисунок 17. Файл JSON для второго запроса

1. Выполнение запроса: Вывести миллиардеров с наибольшим состоянием:

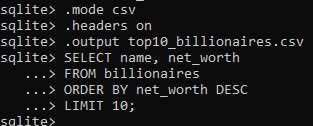


Рисунок 18. Выполнение третьего запроса

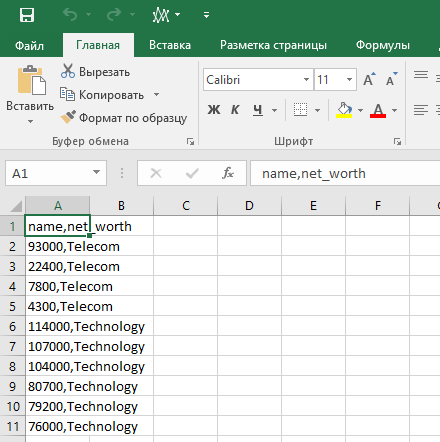


Рисунок 19. Файл CSV для третьего запроса



Рисунок 20. Файл JSON для третьего запроса

1. Выполнение запроса: Вывести среднее состояние миллиардеров для каждой страны:

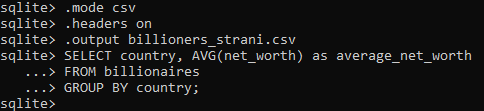


Рисунок 21. Выполнение четвертого запроса

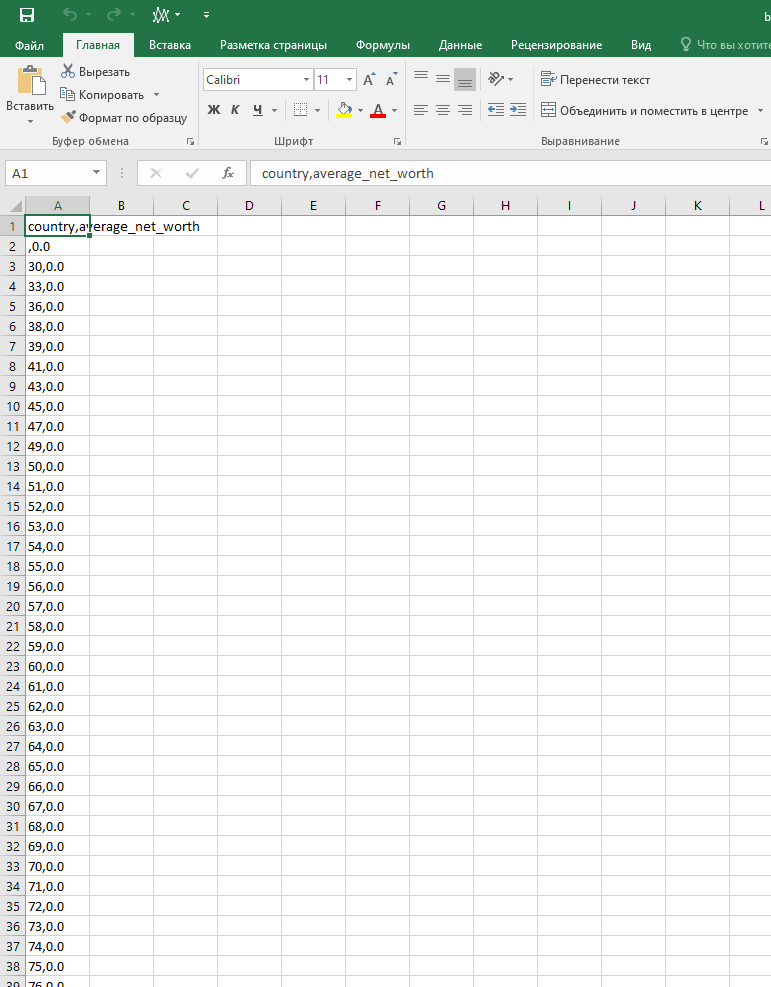


Рисунок 22. Файл CSV для четвертого запроса

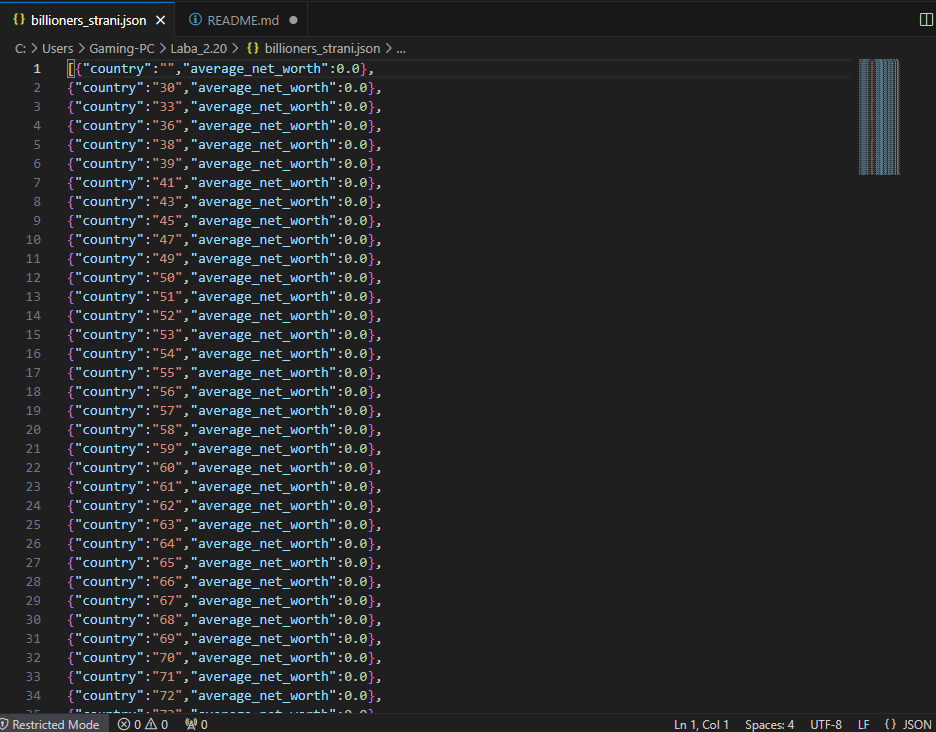


Рисунок 23. Файл JSON для четвертого запроса

1. Выполнение запроса: Вывести миллиардеров с минимальным состоянием:

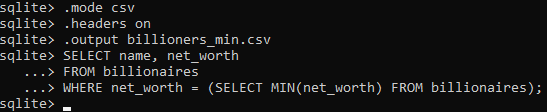


Рисунок 24. Выполнение пятого запроса

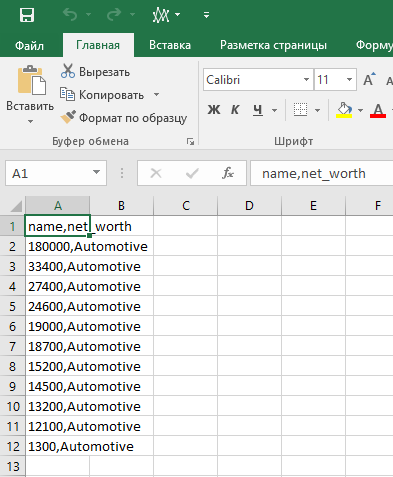


Рисунок 25. Файл CSV для пятого запроса

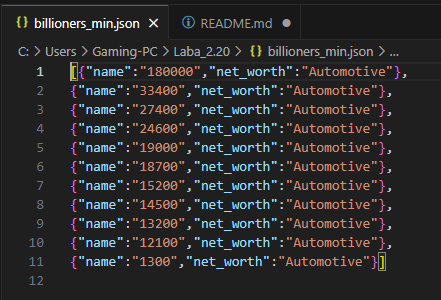


Рисунок 26. Файл JSON пятого запроса

# Задание 10.

После выполнения работы на ветке develop, слила ее с веткой main и отправила изменения на удаленный сервер. Создала файл envirement.yml и деактивировала виртуальное окружение.



Рисунок 27. Деактивация ВО

Ссылка: <https://github.com/Diana-372/Lab2.20.git>

# Ответы на контрольные вопросы:

1. Каково назначение реляционных баз данных и СУБД?

* Реляционные базы данных предназначены для организации и хранения данных в виде таблиц с отношениями между ними.
* Системы управления базами данных (СУБД) предоставляют средства для создания, управления и обращения с базами данных.

# Каково назначение языка SQL?

SQL (Structured Query Language) используется для взаимодействия с реляционными базами данных. Он предоставляет стандартизированный способ создания, изменения, управления и запросов к данным в базе данных.

# Из чего состоит язык SQL?

SQL состоит из нескольких подмножеств:

* DDL (Data Definition Language): Определение структуры базы данных (CREATE, ALTER, DROP).
* DML (Data Manipulation Language): Манипуляция данными (SELECT, INSERT, UPDATE, DELETE).
* DCL (Data Control Language): Управление доступом и правами (GRANT, REVOKE).

# В чем отличие СУБД SQLite от клиент-серверных СУБД?

* SQLite является встроенной базой данных и хранится в виде одного файла.
* Клиент-серверные СУБД (например, MySQL, PostgreSQL) имеют отдельные серверы, к которым подключаются клиенты для доступа к данным.

# Как установить SQLite в Windows и Linux?

* Windows: Можно загрузить исполняемый файл SQLite с официального сайта и выполнить установку.
* Linux: В большинстве дистрибутивов Linux SQLite уже установлен. Для установки можно воспользоваться менеджером пакетов (например, sudo apt-get install sqlite в Ubuntu).

# Как создать базу данных SQLite?

* В командной строке SQLite: sqlite3 имя\_базы\_данных.db.
* Внутри SQLite: CREATE DATABASE имя\_базы\_данных;.

# Как выяснить в SQLite какая база данных является текущей?

* В командной строке SQLite: .database или .dbinfo.
* Внутри SQLite: PRAGMA database\_list;

# Как создать и удалить таблицу в SQLite?

Создание таблицы: CREATE TABLE название ( поле1 тип1,

...

);

Удаление таблицы:

DROP TABLE название;

# Что является первичным ключом в таблице?

Первичный ключ (Primary Key) в таблице — это уникальный идентификатор каждой записи. Он обеспечивает уникальность идентификации записей в таблице.

# Как сделать первичный ключ таблицы автоинкрементным?

При создании таблицы в SQLite можно сделать поле первичного ключа автоинкрементным, используя ключевое слово AUTOINCREMENT

# Каково назначение инструкций NOT и DEFAULT при создании таблиц?

* NOT NULL: Гарантирует, что значение в столбце не может быть NULL (пустым).
* DEFAULT value: Устанавливает значение по умолчанию для столбца, если вставляемые данные не предоставляют значение для этого столбца.

# Каково назначение внешних ключей в таблице? Как создать внешний ключ в таблице?

Внешний ключ (Foreign Key) используется для связи двух таблиц по значениям в столбцах. Он обеспечивает целостность ссылочной целостности данных.

# Как выполнить вставку строки в таблицу базы данных SQLite?

Используйте оператор INSERT INTO.

# Как выбрать данные из таблицы SQLite?

Используйте оператор SELECT.

# Как ограничить выборку данных с помощью условия WHERE?

Используйте WHERE для установки условий: SELECT column1, column2, ...

FROM table\_name WHERE condition;

# Как упорядочить выбранные данные?

Используйте ORDER BY.

# Как выполнить обновление записей в таблице SQLite?

Используйте UPDATE.

# Как удалить записи из таблицы SQLite?

Используйте DELETE.

# Как сгруппировать данные из выборке из таблицы SQLite?

Используйте GROUP BY.

# Как получить значение агрегатной функции (например: минимум, максимум, количество записей и т. д.) в выборке из таблицы SQLite?

Используйте агрегатные функции, такие как MIN, MAX, SUM, AVG, и

т. д.:

# Как выполнить объединение нескольких таблиц в операторе

SELECT?

Используйте JOIN.

# Каково назначение подзапросов и шаблонов при работе с таблицами SQLite?

Подзапросы используются для вложенных запросов, а шаблоны предоставляют средства создания более обобщенных запросов.

# Каково назначение представлений VIEW в SQLite?

Представления позволяют создавать виртуальные таблицы на основе результатов запросов, что облегчает повторное использование и улучшает структуру запросов.

# Какие существуют средства для импорта данных в SQLite?

QLite предоставляет команды .import и .read для импорта данных из внешних источников, а также можно использовать SQL-запросы с оператором INSERT.

# Каково назначение команды .schema ?

Команда .schema используется для вывода SQL-кода, описывающего структуру базы данных, включая определение таблиц, индексов и других объектов

# Как выполняется группировка и сортировка данных в запросах SQLite?

Используйте GROUP BY для объединения строк по значениям в одном или нескольких столбцах.

# Каково назначение "табличных выражений" в SQLite?

Табличные выражения (Common Table Expressions или CTE) позволяют создавать временные результаты запросов, которые можно использовать внутри других запросов. Это обеспечивает более чистый и структурированный код SQL.

# Как осуществляется экспорт данных из SQLite в форматы CSV и JSON?

Экспорт в CSV: Используйте команду .mode csv перед выполнением запроса, а затем .output filename.csv для указания файла вывода.

# Какие еще форматы для экспорта данных Вам известны?

Дополнительно к CSV и JSON, SQLite поддерживает экспорт в форматах XML, HTML, и SQL INSERT. Для экспорта в эти форматы также можно использовать соответствующие команды .mode и .output:

**Вывод:** исследовала базовые возможности системы управления базами данных SQLite3.