

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 РАБОТЕ №2.20
дисциплины «Программирование на языке Python»

Выполнил:
Кондратенко Даниил Витальевич
2 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи, очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: основы работы с SQLite3.

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

Порядок выполнения работы:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

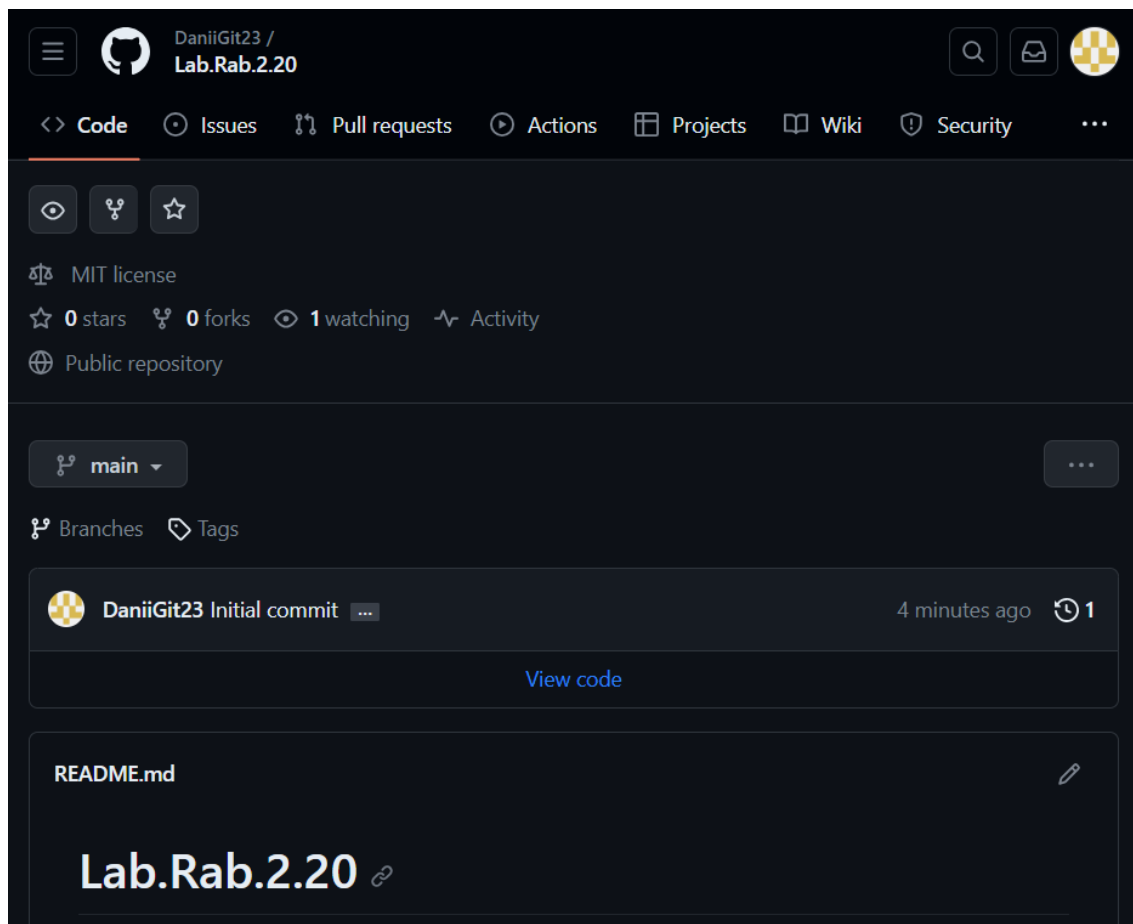


Рисунок 1. Новый репозиторий

Задание 2.

Проклонировал свой репозиторий на свой компьютер.

Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop.

```
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.20.git
Cloning into 'Lab.Rab.2.20'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.20

C:\Users\HUAWEI\Lab.Rab.2.20>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/HUAWEI/Lab.Rab.2.20/.git/hooks]

C:\Users\HUAWEI\Lab.Rab.2.20>
```

Рисунок 2. Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

Задание 3.

Создал виртуальное окружение (ВО) Miniconda и активировал его, также установил необходимые пакеты isort, black, flake8.

```
(base) PS C:\Users\HUAWEI> cd C:\Users\HUAWEI\Lab.Rab.2.20
(base) PS C:\Users\HUAWEI\Lab.Rab.2.20> conda create -n 2.20 python=3.11
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.9.0

Please update conda by running

  $ conda update -n base -c defaults conda
```

Рисунок 3. Создание ВО

```
(2.20) PS C:\Users\HUAWEI\Lab.Rab.2.20> conda install -c conda-forge black
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.9.0

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.9.0
```

Рисунок 4. Установка пакета black

```
(2.20) PS C:\Users\HUAWEI\Lab.Rab.2.20> conda install -c conda-forge flake8
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.9.0

Please update conda by running
```

Рисунок 5. Установка пакета flake8

```
(2.20) PS C:\Users\HUAWEI\Lab.Rab.2.20> conda install -c conda-forge isort
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.5.2
  latest version: 23.9.0
```

Рисунок 6. Установка пакета isort

Пакет isort (isrot) является инструментом для автоматической сортировки импортов в Python-кодах. Он используется для удобства чтения и поддержания порядка в коде.

Пакет black представляет инструмент автоматического форматирования кода для языка Python. Он помогает обеспечить единообразие стиля кодирования в проекте и улучшает читаемость кода.

Пакет flake8 отвечает за статический анализ и проверку Python-кода. Он проводит проверку на соответствие стилю кодирования PEP 8, а также наличие потенциальных ошибок и проблемных паттернов в коде.

Задание 3.

Решить задачу: выполните команды в песочнице:

```
sqlite> create table customer(name);
sqlite> select *
...> from customer
...> select *
...> select * from customer;
Parse error: near "select": syntax error
  select * from customer select * select * from customer;
                        ^--- error here
sqlite> select * from customer;
sqlite> .schema customer
CREATE TABLE customer(name);
sqlite>
```

Рисунок 7. Создание таблицы в SQLite

Что вернула команда .schema?

Команда ".schema customer" вернула схему таблицы "customer".

Задание 4.

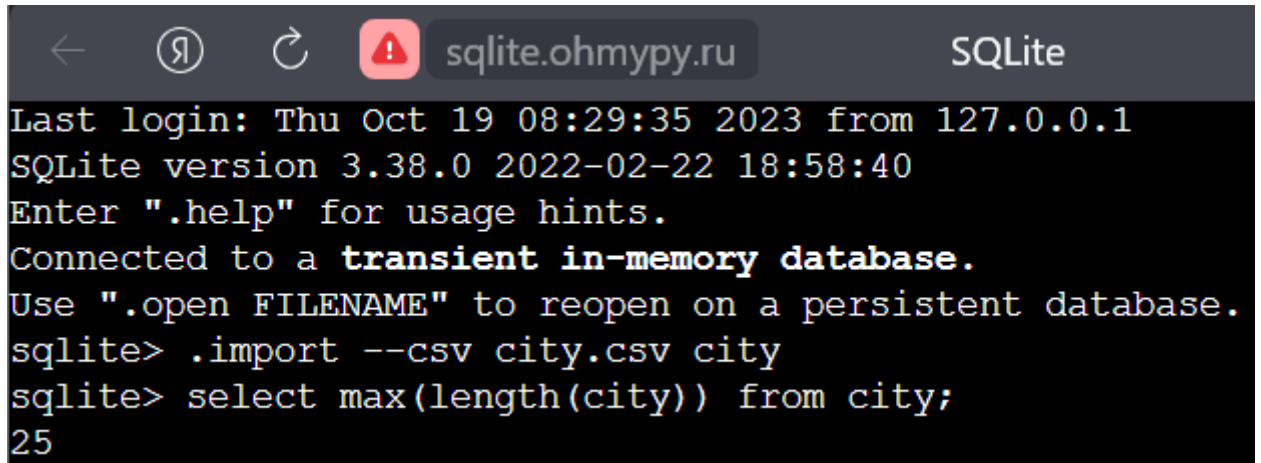
Решите задачу: с помощью команды .help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса.

```
sqlite> create table city(name);
sqlite> .timer on
sqlite> select count(*) from city;
0
Run Time: real 0.000 user 0.000000 sys 0.000155
sqlite>
```

Рисунок 8. Вывод времени выполнения запроса

Задание 5.

Решите задачу: загрузите файл city.csv в песочнице:

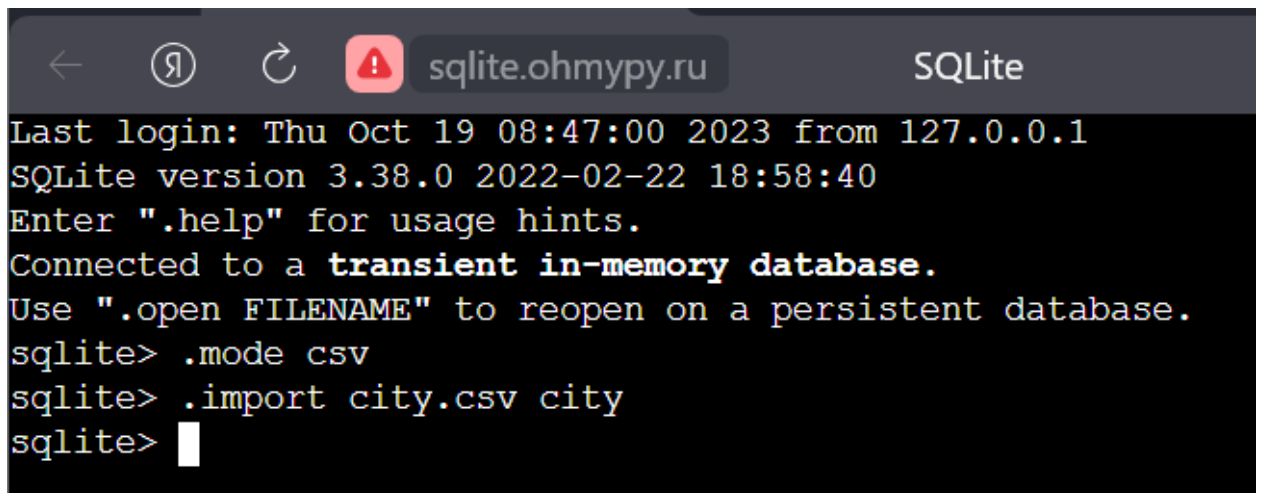


```
← ⓘ ↻ sqlite.ohmypy.ru SQLite
Last login: Thu Oct 19 08:29:35 2023 from 127.0.0.1
SQLite version 3.38.0 2022-02-22 18:58:40
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
```

Рисунок 9. Максимальная длина строки в столбце «city»

Задание 6.

Решите задачу: загрузите файл city.csv в песочнице 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.



```
← ⓘ ↻ sqlite.ohmypy.ru SQLite
Last login: Thu Oct 19 08:47:00 2023 from 127.0.0.1
SQLite version 3.38.0 2022-02-22 18:58:40
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .mode csv
sqlite> .import city.csv city
sqlite> 
```

Рисунок 10. Загрузка файла без использования `--csv`

Задание 7.

Решите задачу: напишите в песочнице запрос, который посчитает количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах. Выведите столбцы `timezone` и `city_count`, отсортируйте по значению часового пояса:

```
sqlite> .import --csv city.csv city
sqlite> SELECT timezone, COUNT(*) AS city_count
...> FROM city
...> WHERE federal_district IN ('Сибирский', 'Приволжский')
...> GROUP BY timezone
...> ORDER BY timezone;
UTC+3|101
UTC+4|41
UTC+5|58
UTC+6|6
UTC+7|86
UTC+8|22
sqlite> 
```

Рисунок 11. Подсчет количества городов

Задание 8.

Напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

```
sqlite> .import --csv city.csv city
sqlite> SELECT timezone, COUNT(*) AS city_count
...> from city
...> GROUP BY timezone
...> ORDER BY city_count DESC;
UTC+3|660
UTC+5|173
UTC+7|86
UTC+4|66
UTC+9|31
UTC+8|28
UTC+2|22
UTC+10|22
UTC+11|17
UTC+6|6
UTC+12|6
sqlite> 
```

Рисунок 12. Обычный запрос

Сделать такой же запрос только в формате CSV с заголовками и разделителем «|».

```
sqlite> .import --csv city.csv city
sqlite> .headers on
sqlite> .mode csv
sqlite> .separator |
sqlite> SELECT timezone, COUNT(*) AS city_count
...> FROM city
...> GROUP BY timezone
...> ORDER BY city_count DESC;
timezone|city_count
UTC+3|660
UTC+5|173
UTC+7|86
UTC+4|66
UTC+9|31
UTC+8|28
UTC+2|22
UTC+10|22
UTC+11|17
UTC+6|6
UTC+12|6
sqlite> 
```

Рисунок 12. Запрос в формате CSV

Задание 9.

Индивидуальное задание: загрузите в SQLite выбранный Вами датасет в формате CSV (датасет можно найти на сайте Kaggle). Сформируйте более пяти запросов к таблицам БД. Выгрузите результат выполнения запросов в форматы CSV и JSON.

- 1) Используем команду для запроса вывода всех строк и столбцов из таблицы `SELECT * FROM mobile;`

```
sqlite> .import --csv mobile.csv mobile
sqlite> SELECT * FROM mobile;
Samsung|Galaxy S10|yes|3 Cameras: 12, 12, 16 MP|1440 x 3040|AMOLED|Single|Android|White|Greater Accra|Circle|6.1|3400.0|128.0|8.0|10.0|2450.0
Samsung|Galaxy Note 20 Ultra|yes|3 Cameras: 108, 12, 12 MP|1440 x 3088|AMOLED|Dual|Android|Bronze|Greater Accra|Circle|6.9|4500.0|128.0|12.0|10.0|5000.0
Samsung|Galaxy A54 5G|yes|3 Cameras: 50, 12, 5 MP|1080 x 2340|Super AMOLED|Dual|Android|Black|Greater Accra|Accra Metropolitan|6.4|5000.0|256.0|8.0|32.0|3000.0
Tecno|Camon 20 Pro|no|2 Cameras: 64, 2 MP|1080 x 2400|AMOLED|Dual|Android|Black|Greater Accra|Accra Metropolitan|6.67|5000.0|256.0|8.0|32.0|1900.0
Samsung|Galaxy S22 Ultra|no|4 Cameras: 108, 10, 10, 12 MP|1400 x 3088|AMOLED|Dual|Android|Black|Greater Accra|Accra Metropolitan|6.8|5000.0|512.0|12.0|40.0|5750.0
Samsung|Galaxy A73 5G|yes|4 Cameras: 108, 12, 5, 5 MP|1080 x 2400|Super AMOLED|Dual|Android|Black|Greater Accra|Accra Metropolitan|6.7|5000.0|256.0|8.0|32.0|2555.0
Samsung|Galaxy A54 5G|yes|3 Cameras: 50, 12, 5 MP|1080 x 2340|Super AMOLED|Dual|Android|Blue|Greater Accra|Accra Metropolitan|6.4|5000.0|256.0|8.0|32.0|2800.0
Samsung|Galaxy S23 Ultra|no|4 Cameras: 200, 10, 10, 12 MP|1440 x 3088|AMOLED|Dual|Android|Green|Greater Accra|Accra Metropolitan|6.8|5000.0|512.0|12.0|12.0|8480.0
Samsung|Galaxy A54 5G|yes|3 Cameras: 50, 12, 5 MP|1080 x 2340|Super AMOLED|Dual|Android|White|Greater Accra|Accra Metropolitan|6.4|5000.0|256.0|8.0|32.0|3710.0
Samsung|Galaxy A54 5G|yes|3 Cameras: 50, 12, 5 MP|1080 x 2340|Super AMOLED|Dual|Android|Black|Greater Accra|Accra Metropolitan|6.4|5000.0|256.0|8.0|32.0|3710.0
Tecno|Camon 20 Premier 5G|yes|3 Cameras: 50, 108, 2 MP|1080 x 2400|AMOLED|Dual|Android|Blue|Greater Accra|Accra Metropolitan|6.67|5000.0|512.0|8.0|32.0|2440.0
```

Рисунок 13. Вывод всех строк и столбцов файла

- 2) Вывести только определенного столбца из таблицы.

```
sqlite> .import --csv mobile.csv mobile
sqlite> SELECT brand FROM mobile;
Samsung
Samsung
Samsung
Tecno
Samsung
Samsung
Samsung
Samsung
Samsung
Samsung
Tecno
Tecno
Samsung
Samsung
Samsung
Samsung
Tecno
Tecno
Samsung
Tecno
Tecno
```

Рисунок 14. Определенный столбец таблицы

3) Подсчет числа строк в таблице.

```
sqlite> SELECT count(*) FROM mobile;  
3600  
sqlite>
```

Рисунок 15. Подсчет числа строк

4) Подсчет числа строк для каждого уникального значения в столбце.

```
sqlite> SELECT main_camera, count(*) FROM mobile GROUP BY main_camera;  
|235  
1 Cameras: 008 MP|1  
1 Cameras: 02 MP|1  
1 Cameras: 03 MP|5  
1 Cameras: 12 MP|611  
1 Cameras: 122 MP|24  
1 Cameras: 13 MP|86  
1 Cameras: 16 MP|106  
1 Cameras: 1966 MP|5  
1 Cameras: 2 MP|9  
1 Cameras: 4 MP|1  
1 Cameras: 48 MP|46  
1 Cameras: 5 MP|14  
1 Cameras: 50 MP|36  
1 Cameras: 64 MP|28  
1 Cameras: 8 MP|65  
2 Cameras: 108, 8 MP|1  
2 Cameras: 12, 12 MP|399  
2 Cameras: 12, 16 MP|41  
2 Cameras: 12, 2 MP|1  
2 Cameras: 12, 20 MP|12  
2 Cameras: 122, 12 MP|27  
2 Cameras: 122, 16 MP|30  
2 Cameras: 122, 8 MP|12  
2 Cameras: 13, 01 MP|3
```

Рисунок 16. Подсчет числа строк для каждого уникального значения

5) Максимальная длина строки в столбце «model»

```
sqlite> select max(length(model)) from mobile;  
24
```

Рисунок 17. Максимальная длина строки

Сохраним результаты запросов в форматы CSV и JSON, используя команду ".output".

Слил ветку develop с веткой main

```
C:\Users\HUAWEI\Lab.Rab.2.20>git merge develop
Updating 4e87a58..47aa6bc
Fast-forward
 .gitignore      |    2 +-
 1.sql           |    6 +
 2.sql           |    5 +
 3.sql           |    3 +
 4.sql           |    2 +
 5.sql           |   13 +
 6.1.sql         |   17 +
 6.2.sql         |   21 +
 IDZ.csv         | 3600 ++++++
 IDZ.json        |    0
 README.md       |    3 +-
 environment.yml | Bin 0 -> 2214 bytes
 mobile.csv      | 3601 ++++++
 13 files changed, 7271 insertions(+), 2 deletions(-)
```

Рисунок 10. Слияние веток

Создал файл с зависимостями и деактивировал ВО.

```
(2.20) PS C:\Users\HUAWEI\Lab.Rab.2.20> conda deactivate
(base) PS C:\Users\HUAWEI\Lab.Rab.2.20> |
```

Рисунок 11. Деактивация ВО

Ссылка на репозиторий: <https://github.com/DaniiGit23/Lab.Rab.2.20>

Ответы на контрольные вопросы:

1. Назначение реляционных баз данных и СУБД состоит в организации и хранении структурированных данных, а также обеспечении возможности эффективного доступа и манипуляции с этими данными.

2. Назначение языка SQL (Structured Query Language) заключается в обеспечении возможности выполнения операций с данными в реляционных базах данных. Он используется для создания таблиц, вставки, обновления и удаления данных, а также для выполнения запросов и анализа данных.

3. Язык SQL состоит из различных типов операторов, таких как операторы создания таблиц, операторы вставки данных, операторы выборки данных и операторы обновления данных, а также операторы удаления данных.

4. Отличие СУБД SQLite от клиент-серверных СУБД заключается в том, что SQLite является файловой СУБД, то есть база данных хранится в одном файле на компьютере или мобильном устройстве, тогда как клиент-серверные СУБД используют клиентское и серверное приложения для взаимодействия с базой данных, которая хранится на сервере.

5. Чтобы установить SQLite в Windows, необходимо скачать установочный файл с официального сайта SQLite, запустить его и следовать инструкциям установщика. В Linux SQLite обычно уже установлен вместе с операционной системой.

6. Чтобы создать базу данных SQLite, можно использовать команду "CREATE DATABASE имя_базы_данных;" в инструменте командной строки или специальной программе для работы с SQLite.

7. Чтобы выяснить, какая база данных является текущей в SQLite, можно использовать команду "PRAGMA database_list;", которая покажет список всех баз данных и пометит текущую базу данных.

8. Чтобы создать таблицу в SQLite, можно использовать команду "CREATE TABLE имя_таблицы (столбец1 тип1, столбец2 тип2, ...);" в инструменте командной строки или специальной программе для работы с SQLite. Чтобы удалить таблицу, используется команда "DROP TABLE имя_таблицы;".

9. Первичный ключ в таблице является уникальным идентификатором каждой записи в таблице. Он уникально идентифицирует каждую запись и используется для обеспечения ссылочной целостности в базе данных.

10. Чтобы сделать первичный ключ таблицы автоинкрементным в SQLite, необходимо использовать тип данных INTEGER PRIMARY KEY AUTOINCREMENT при определении столбца первичного ключа.

11. Инструкция NOT используется при создании таблицы для указания, что столбец не может содержать значение NULL (нулевое значение). DEFAULT используется для указания значения по умолчанию для столбца, если вставляется запись без значения для этого столбца.

12. Внешние ключи в таблице служат для установления связи между таблицами. Они указывают на первичный ключ в другой таблице. Чтобы создать внешний ключ в таблице, используется команда "FOREIGN KEY (столбец) REFERENCES имя_таблицы(столбец);".

13. Чтобы выполнить вставку строки в таблицу базы данных SQLite, можно использовать команду "INSERT INTO имя_таблицы (столбец1, столбец2, ...) VALUES (значение1, значение2, ...);".

14. Чтобы выбрать данные из таблицы SQLite, можно использовать команду "SELECT столбец1, столбец2, ... FROM имя_таблицы;".

15. Чтобы ограничить выборку данных с помощью условия WHERE, нужно использовать команду "SELECT * FROM имя_таблицы WHERE условие;".

16. Чтобы упорядочить выбранные данные, используется команда "SELECT * FROM имя_таблицы ORDER BY столбец ASC/DESC;", где ASC указывает на сортировку по возрастанию, а DESC - по убыванию.

17. Чтобы выполнить обновление записей в таблице SQLite, используется команда "UPDATE имя_таблицы SET столбец1 = новое_значение1, столбец2 = новое_значение2 WHERE условие;".

18. Чтобы удалить записи из таблицы SQLite, используется команда "DELETE FROM имя_таблицы WHERE условие;".

19. Чтобы сгруппировать данные из выборки из таблицы SQLite, используется команда "SELECT столбец, агрегатная_функция(столбец) FROM имя_таблицы GROUP BY столбец;".

20. Чтобы получить значение агрегатной функции в выборке из таблицы SQLite, используется команда "SELECT агрегатная_функция(столбец) FROM имя_таблицы;".

21. Чтобы выполнить объединение нескольких таблиц в операторе SELECT, используется команда "SELECT * FROM таблица1 JOIN таблица2 ON таблица1.столбец1 = таблица2.столбец2;".

22. Подзапросы используются в SQLite для выполнения запросов внутри других запросов. Шаблоны служат для определения конкретных значений в запросах на основе паттернов или шаблонов.

23. Представления VIEW в SQLite используются для создания виртуальных таблиц на основе существующих таблиц. Они предоставляют удобный способ абстрагирования и доступа к данным из нескольких таблиц.

24. Для импорта данных в SQLite можно использовать различные средства, например, команду "INSERT INTO имя_таблицы SELECT * FROM имя_внешней_таблицы;" или специальные программы для импорта данных из разных форматов.

25. Команда ".schema" в SQLite используется для отображения схемы базы данных, включая определения таблиц, индексов и других объектов базы данных.

26. Группировка и сортировка данных в запросах SQLite осуществляется с помощью команды "GROUP BY" и "ORDER BY" соответственно.

27. Табличные выражения в SQLite используются для создания временных таблиц внутри операторов SELECT, что позволяет вычислять и структурировать данные перед выполнением запросов.

28. Для экспорта данных из SQLite в форматы CSV (Comma-Separated Values) и JSON (JavaScript Object Notation) можно использовать различные инструменты и программы, включая SQLite Shell и специальные библиотеки.

29. Кроме CSV и JSON, для экспорта данных SQLite можно использовать и другие форматы, такие как XML (eXtensible Markup Language) или Excel (в формате XLSX).

Вывод: в ходе выполнения работы исследовали базовые возможности системы управления базами данных SQLite3.