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 «Анализ данных»
Вариант 13

Выполнил:
Иващенко Олег Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.02 «Информационные и вычислительные машины»,
направленность (профиль)
«Программное обеспечение средств вычислительной техники и автоматизированных систем»

_____
(подпись)

Руководитель практики:
Воронкин Роман Александрович,
доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2024 г.

Тема: «Основы работы с SQLite3»

Цель: Исследовать базовые возможности системы управления базами данных SQLite3.

Порядок выполнения работы

```
sqlite> CREATE TABLE customer(name);
sqlite> SELECT * FROM customer;
sqlite> .schema customer
CREATE TABLE customer(name);
sqlite> .schema customer;
```

Рисунок 1.1 – Выполнение создание таблицы

SQLite позволяет отслеживать время выполнения запроса с помощью команды .timer on/off.

```
sqlite> .timer on
sqlite> SELECT COUNT(*) FROM city;
Run Time: real 0.001 user 0.000000 sys 0.000512
Parse error: no such table: city
```

Рисунок 1.2 – Включение таймера выполнения операции

```
sqlite> .import --csv city.csv city
sqlite> SELECT MAX(length(city)) FROM city;
25
```

Рисунок 1.3 – Выполнение задания №9

```
sqlite> .mode csv
sqlite> .import city.csv city
sqlite> SELECT COUNT(city) FROM city;
1117
```

Рисунок 1.4 – Импорт CSV-файла без использования ключа –csv

Задача: Написать SQL-запрос, который посчитает количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах. Вывести столбцы timezone и city_count, отсортировав по значению часового пояса.

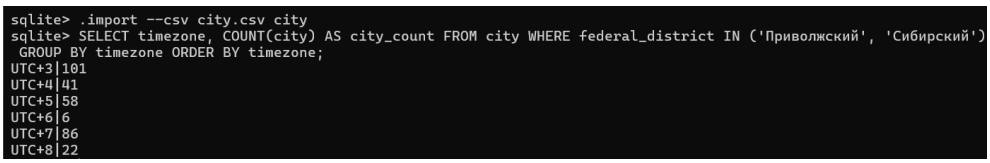


Рисунок 1.5 – Вывод часовых поясов в Сибирском и Приволжском федеральных округах с соответствующим им количеством городов

Задача: Написать SQL-запрос, который найдёт трёх ближайших к Самаре города, не считая саму Самару. В ответе указать названия этих городов через запятую в порядке удаления от Самары.

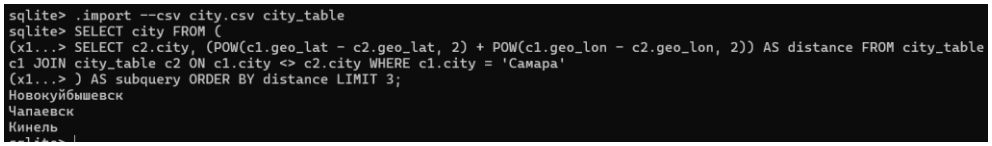


Рисунок 1.6 – Вывод трёх ближайших городов к городу Самара

Задача: Написать SQL-запрос, который посчитает количество городов в каждом часовом поясе. Отсортировать по количеству городов по убыванию. Также выполнить запрос, чтобы результат был в формате CSV, с заголовками и с разделителями «pipe» (|).

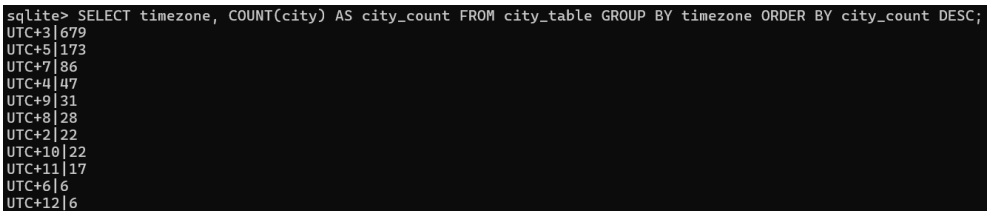


Рисунок 1.7 – Вывод часовых поясов и соответствующее им количество городов в порядке убывания

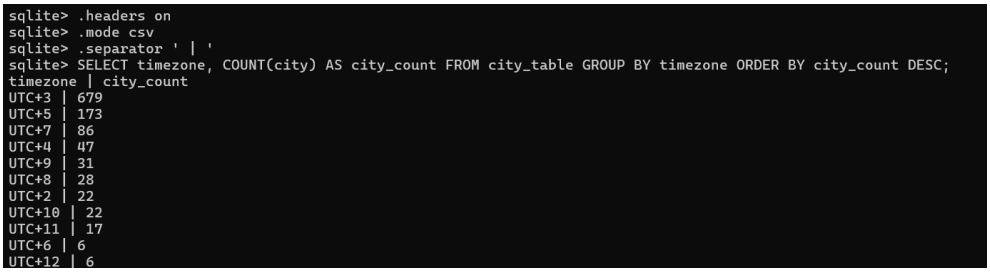
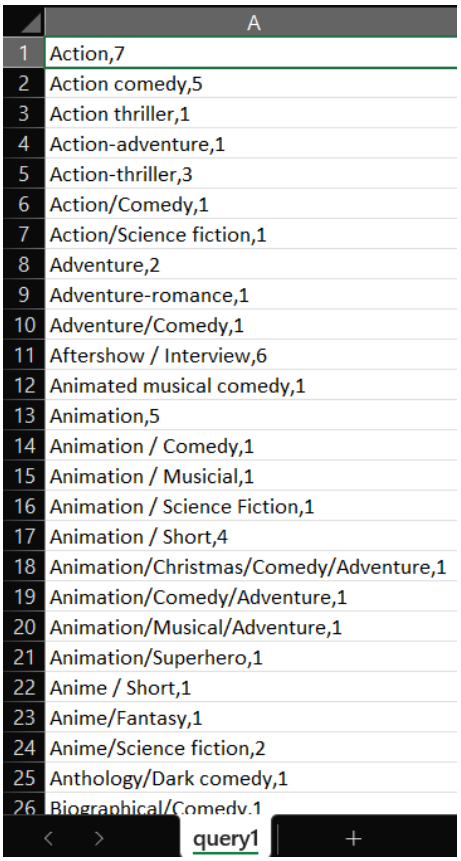


Рисунок 1.8 – Вывод часовых поясов и соответствующее им количество городов в порядке убывания с заголовками, в формате csv и разделителем «|»

Индивидуальное задание. Загрузить выбранный датасет в формате CSV. Сформировать более пяти запросов к таблицам БД. Выгрузить результат выполнения запросов в форматы CSV и JSON.

```
sqlite> .mode csv
sqlite> .import netflix.csv netflix
sqlite> .output query1.csv
sqlite> SELECT genre, COUNT(*) AS count FROM netflix GROUP BY genre;
sqlite>
sqlite> .mode json
sqlite> .output query1.json
sqlite> SELECT genre, COUNT(*) AS count FROM netflix GROUP BY genre;
```

Рисунок 2.1.1 – Вывод жанров и соответствующее количество фильмов



	A
1	Action,7
2	Action comedy,5
3	Action thriller,1
4	Action-adventure,1
5	Action-thriller,3
6	Action/Comedy,1
7	Action/Science fiction,1
8	Adventure,2
9	Adventure-romance,1
10	Adventure/Comedy,1
11	Aftershow / Interview,6
12	Animated musical comedy,1
13	Animation,5
14	Animation / Comedy,1
15	Animation / Musicial,1
16	Animation / Science Fiction,1
17	Animation / Short,4
18	Animation/Christmas/Comedy/Adventure,1
19	Animation/Comedy/Adventure,1
20	Animation/Musical/Adventure,1
21	Animation/Superhero,1
22	Anime / Short,1
23	Anime/Fantasy,1
24	Anime/Science fiction,2
25	Anthology/Dark comedy,1
26	Biographical/Comedy,1

query1

Рисунок 2.1.2 – Вывод результатов в CSV

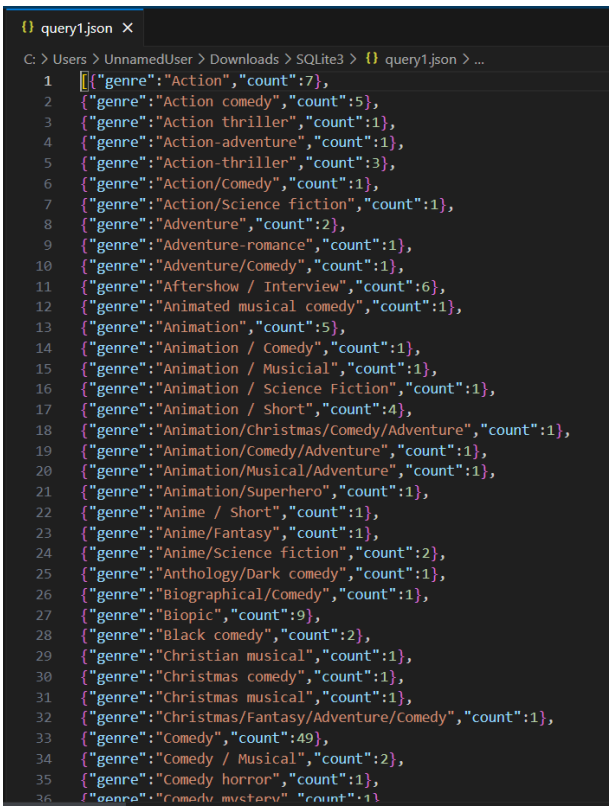


Рисунок 2.1.3 – Вывод результатов в JSON

```
sqlite> .mode csv
sqlite> .output query2.csv
sqlite> SELECT title, year FROM netflix WHERE year > 2017 ORDER BY year;
sqlite>
sqlite> .mode json
sqlite> .output query2.json
sqlite> SELECT title, year FROM netflix WHERE year > 2017 ORDER BY year;
```

Рисунок 2.2.1 – Вывод всех фильмов, вышедших после 2017 года (не включительно)

Рисунок 2.2.2 – Вывод результатов в CSV

```
[{"title":"To Each, Her Own","year":"2018"},
{"title":"Happy Anniversary","year":"2018"},
{"title":"Out of Many, One","year":"2018"},
{"title":"Game Over, Man!","year":"2018"},
{"title":"Forgive Us Our Debts","year":"2018"},
{"title":"Apostle","year":"2018"},
{"title":"The Polka King","year":"2018"},
{"title":"Paradox","year":"2018"},
{"title":"The American Meme","year":"2018"},
{"title":"Irreplaceable You","year":"2018"},
{"title":"The Princess Switch","year":"2018"},
{"title":"ReMastered: Who Killed Jam Master Jay?","year":"2018"},
{"title":"Seeing Allred","year":"2018"},
{"title":"Springsteen on Broadway","year":"2018"},
{"title":"Angela's Christmas","year":"2018"},
{"title":"When We First Met","year":"2018"},
{"title":"Alex Strangelove","year":"2018"},
{"title":"How It Ends","year":"2018"},
{"title":"End Game","year":"2018"},
{"title":"I Am Not an Easy Man","year":"2018"},
{"title":"The Angel","year":"2018"},
{"title":"Porta dos Fundos: The Last Hangover","year":"2018"},
{"title":"Lust Stories","year":"2018"},
{"title":"Step Sisters","year":"2018"},
{"title":"The Most Assassinated Woman in the World","year":"2018"},
{"title":"Recovery Boys","year":"2018"},
{"title":"Rajma Chawal","year":"2018"},
{"title":"The Kissing Booth","year":"2018"},
{"title":"The Trader","year":"2018"},
{"title":"Benji","year":"2018"},
{"title":"5 Star Christmas","year":"2018"},
{"title":"Ladies First","year":"2018"},
{"title":"Calibre","year":"2018"},
{"title":"To All the Boys I've Loved Before","year":"2018"},
{"title":"Take Your Pills","year":"2018"},
{"title":"Two Catalonias","year":"2018"}
```


Рисунок 2.2.3 – Вывод результатов в JSON

```
sqlite> .mode csv
sqlite> .output query3.csv
sqlite> SELECT title, imdb_score FROM netflix WHERE (imdb_score > 8 AND genre='Documentary') ORDER BY imdb_score DESC;
sqlite>
sqlite> .mode json
sqlite> .output query3.json
sqlite> SELECT title, imdb_score FROM netflix WHERE (imdb_score > 8 AND genre='Documentary') ORDER BY imdb_score DESC;
```

Рисунок 2.3.1 – Вывод всех фильмов жанра «Документальный» с оценкой выше 8

A
David Attenborough: A Life on Our Planet,9
Emicida: AmarElo - It's All For Yesterday,8.6
Winter on Fire: Ukraine's Fight for Freedom,8.4
Cuba and the Cameraman ,8.3
Dancing with the Birds,8.3
13th,8.2
Seaspiracy,8.2
Disclosure: Trans Lives on Screen,8.2
The Three Deaths of Marisela Escobedo,8.2
My Octopus Teacher,8.1
Chasing Coral ,8.1
Rising Phoenix,8.1

Рисунок 2.3.2 – Вывод результатов в CSV

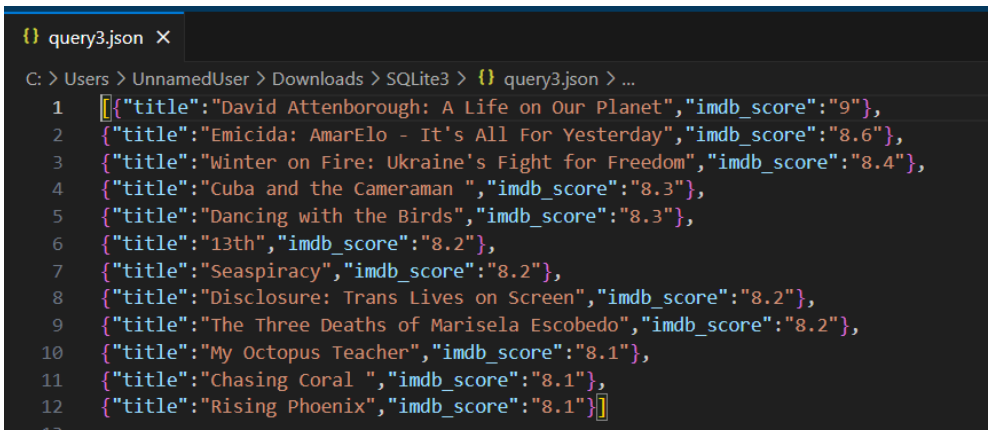


Рисунок 2.3.3 – Вывод результатов в JSON

```
sqlite> .mode csv
sqlite> .output query4.csv
sqlite> .separator ' | '
sqlite> SELECT title, genre, imdb_score FROM netflix ORDER BY imdb_score DESC LIMIT 10;
sqlite>
sqlite> .mode json
sqlite> .output query4.json
sqlite> SELECT title, genre, imdb_score FROM netflix ORDER BY imdb_score DESC LIMIT 10;
```

Рисунок 2.4.1 – Вывод топ-10 фильмов с жанром по рейтингу

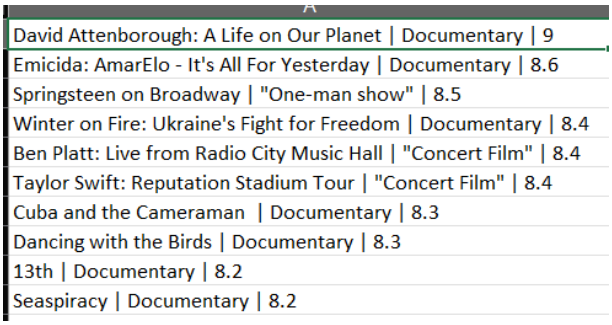


Рисунок 2.4.2 – Вывод результатов в CSV

Рисунок 2.4.3 – Вывод результатов в JSON

```
sqlite> .mode csv
sqlite> .output query5.csv
sqlite> .separator ' | '
sqlite> SELECT genre, AVG(runtime) AS avgtime FROM netflix GROUP BY genre ORDER BY avgtime DESC;
sqlite>
sqlite> .mode json
sqlite> .output query5.json
sqlite> SELECT genre, AVG(runtime) AS avgtime FROM netflix GROUP BY genre ORDER BY avgtime DESC;
```

Рисунок 2.5.1 – Вывод всех жанров и их средней продолжительности в порядке убывания

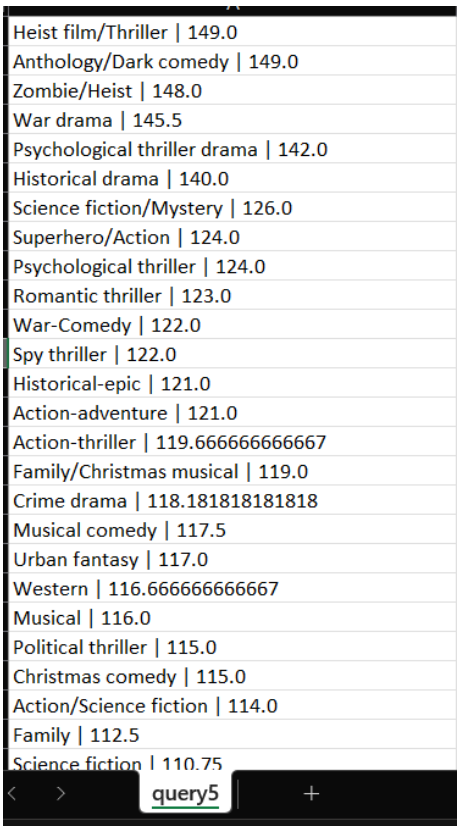


Рисунок 2.5.2 – Вывод результатов в CSV

Рисунок 2.5.3 – Вывод результатов в JSON

Контрольные вопросы

1. Каково назначение реляционных баз данных и СУБД?

Реляционные БД с СУБД предназначены для хранения, управления и извлечения структурированных данных. Они обеспечивают эффективный и удобный способ работы с данными.

2. Каково назначение языка SQL?

SQL (Structured Query Language) предназначен для управления данными в реляционных базах данных. Он используется для создания и управления БД, выполнения запросов, вставки, обновления и удаления данных, а также для управления безопасностью и целостностью данных.

3. Из чего состоит язык SQL?

SQL состоит из четырёх основных типов команд:

- DDL (Data Definition Language) – используется для определения структуры БД, такой как создание, изменение или удаление объектов БД (например, CRETE, ALERT, DROP).
- DML (Data Manipulation Language) – используется для работы с данными в БД, таких как добавление, изменение, удаление и выборка данных (INSERT, UPDATE, DELETE, SELECT).
- DCL (Data Control Language) – используется для управления правами доступа к данными (например, GRANT, REVOKE).
- TCL (Transaction Control Language) – используется для управления транзакциями в БД (например, COMMIT, ROLLBACK).

4. В чём отличие СУБД SQLite от клиент-серверных СУБД?

В отличие от клиент-серверных СУБД, SQLite является встраиваемой БД, которая не требует отдельного сервера. Он работает непосредственно с файлами на диске. Клиент-серверных СУБД, напротив, используют клиент-

серверную архитектуру, где сервер управляет БД, а клиенты подключаются к серверу для доступа к данным.

5. Как установить SQLite в Windows и Linux?

Для установки SQLite в Windows можно скачать предварительно собранные бинарные файлы SQLite с официального сайта. После скачивания файлы нужно разархивировать, после чего можно начинать работу.

В большинстве дистрибутивов Linux SQLite уже предустановлен. Если нет, можно установить его через менеджер пакетов дистрибутива (sudo apt-get install sqlite3).

6. Как создать базу данных SQLite?

Для создания БД в SQLite можно использовать команду:

CREATE DATABASE <db_name>

7. Как выяснить в SQLite какая база данных является текущей?

В SQLite не существует концепции «текущей» БД. При работе с SQLite указывается только имя файла БД, с которой ведётся работа.

8. Как создать и удалить таблицу в SQLite?

Для создания таблицы используется команда CREATE TABLE <table_name> (column1 datatype1, column2 datatype2…);

Для удаления таблицы используется команда DROP TABLE <table_name>.

9. Что является первичным ключом в таблице?

Первичный ключ (Primary Key) – это уникальный идентификатор каждой записи в таблице.

10. Как сделать первичный ключ таблицы автоинкрементным?

Для того, чтобы сделать первичный ключ автоинкрементным, используется ключевое слово AUTOINCREMENT:
CREATE TABLE <table_name> (id INTEGER PRIMARY KEY AUTOINCREMENT, column1 datatype1, …);

11. Каково назначение инструкций NOT NULL и DEFAULT при создании таблиц?

NOT NULL используется для указания того, что значение столбца не может быть NULL (пустым).
DEFAULT используется для установки значения по умолчанию для столбца.

12. Каково назначение внешних ключей в таблице? Как создать внешний ключ в таблице?

Внешние ключи используются для связи строк в одной таблице со строками в другой таблице. Внешний ключ определяется на столбце, который ссылается на первичный ключ в другой таблице. Пример применения:
CREATE TABLE <table_name> (
orderID INTEGER PRIMARY KEY,
productID INTEGER,
FOREIGN KEY (productID) REFERENCES products(productID)

13. Как выполнить вставку строки в таблицу базы данных SQLite?

Для вставки строки в таблице SQLite используется следующий синтаксис:
INSERT INTO <table_name> (column1, column2, …) VALUES (value1, value2, …);

14. Как выбрать данные из таблицы SQLite?

Для выборки используется следующий синтакстис:
SELECT column1, column2, … FROM <table_name>;

15. Как ограничить выборку данных с помощью условия WHERE?

SELECT * FROM <table_name> WHERE <условие>;

16. Как упорядочить выбранные данные?

Упорядочивание в SQL осуществляется с помощью параметра ORDER BY <column> [DESC].

17. Как выполнить обновление записей в таблице SQLite?

UPDATE <table_name> SET column1 = value1, column2 = value2, … WHERE <условие>;

18. Как удалить записи из таблицы SQLite?

DELETE FROM <table_name> WHERE <условие>;

19. Как сгруппировать данные из выборки из таблицы SQLite?

SELECT * FROM <table_name> GROUP BY <column>;

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

SELECT <function_name>(<column>) AS <name> FROM <table_name>;

21. Как выполнить объединение нескольких таблиц в операторе SELECT?

SELECT column1, column2, … FROM <table_name> JOIN <table2_name> ON <условие>;

22. Каково назначение подзапросов и шаблонов при работе с таблицами SQLite?

Подзапросы используются для выполнения запроса внутри другого запроса. Это позволяет создавать более сложные запросы, комбинируя результаты из нескольких запросов.

23. Каково назначение представлений VIEW в SQLite?

Представления (VIEW) в SQLite используются для создания виртуальных таблиц, которые можно использовать для упрощения запросов. Они сохраняются в БД и могут быть использованы повторно.

24. Какие существуют средства для импорта данных в SQLite?

Данные могут быть импортированы в SQLite различными способами:

- Используя команду импорта .import в консоли SQLite;
- Используя командную строку SQLite с параметрами, чтобы выполнить импорт из файла;
- Используя инструменты для работы с БД, такие как SQLiteStudio, чтобы импортировать данные из различных форматов файлов (CSV, JSON и т.д.).

25. Каково назначение команды .schema?

Команда .schema используется для отображения схемы (структуры) таблицы или БД SQLite. Она показывает SQL-код, который используется для создания таблицы или БД.

26. Как выполняется группировка и сортировка данных в запросах SQLite?

SELECT column1, column2, … FROM <table_name> GROUP BY <column> ORDER BY <column> [ASC/DESC];

27. Каково назначение «табличных выражений» в SQLite?

Табличные выражения позволяют использовать вывод запроса как временную таблицу в других частях запроса. Они полезны, когда необходимо выполнить запрос на основе результата другого запроса. Например:

SELECT * FROM (SELECT column1, column2 FROM <table_name>) AS temp_table;

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

Импорт данных в файл формата CSV:

.mode csv

.output <file_name>.csv

SELECT * FROM <table_name>;

Импорт данных в файл формата JSON:

.mode json

.output <file_name>.json

SELECT * FROM <table_name>;

29. Какие ещё форматы для экспорта данных Вам известны?

Помимо CSV и JSON данные также могут быть экспортированы в XML, SQL, HTML и другие форматы с помощью инструментов и расширений.

Выводы: В процессе выполнения лабораторной работы были исследованы базовые возможности системы управления базами данных SQLite3, выполнены задания и индивидуальное задание.