

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 №6**  
**дисциплины «Анализ данных»**  
**Вариант 9**

Выполнил:  
Дудкин Константин Александрович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»  
направление «Программное  
обеспечение средств вычислительной  
техники и автоматизированных  
систем»,  
очная форма обучения

---

(подпись)

Руководитель практики:  
Кандидат технических наук, доцент  
кафедры инфокоммуникаций, доцент  
Воронкин Роман Александрович

---

(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

Решил задачи:

```
sqlite> create table customer(name);  
sqlite> select *  
...> from customer;  
sqlite> .schema customer  
CREATE TABLE customer(name);  
sqlite> █
```

Рисунок 1. Задача 1

```
sqlite> .timer on  
sqlite> select count(*) from city;  
1117  
Run Time: real 0.000 user 0.000236 sys 0.000000  
█
```

Рисунок 2. Задача 2. Здесь нужно было использовать .timer on

```
sqlite> █select max(length(city)) from city;  
25  
Run Time: real 0.001 user 0.000597 sys 0.000477  
█
```

Рисунок 3. Задача 3

```
sqlite> █.separator ,  
sqlite> .import city.csv city;  
█
```

Рисунок 4. Задача 4

```

sqlite> .mode box
sqlite> SELECT
...> DISTINCT timezone AS time,
...> COUNT() AS count
...> FROM city
...> GROUP BY time
...> ORDER BY time DESC;

```

time	count
UTC+9	31
UTC+8	28
UTC+7	86
UTC+6	6
UTC+5	173
UTC+4	66
UTC+3	660
UTC+2	22
UTC+12	6
UTC+11	17
UTC+10	22

```

Run Time: real 0.003 user 0.002024 sys 0.000000
sqlite>

```

Рисунок 5. Задача 5

```

sqlite> WITH samara AS (SELECT geo_lat AS lat, geo_lon AS lon FROM city WHERE city="Самара")
...> SELECT
...> city,
...> SQRT(
...> POW (geo_lon - lon, 2) + POWER(geo_lat - lat, 2)
...> ) * 69.09 distance
...> FROM
...> city, samara
...> WHERE distance > 0
...> ORDER BY distance
...> LIMIT 3;

```

city	distance
Новокуйбышевск	12.8298063265514
Чапаевск	24.7389598092284
Кинель	36.4840951529617

```

Run Time: real 0.016 user 0.009177 sys 0.000581
sqlite>

```

Рисунок 6. Задача 6

```

sqlite> .mode csv
sqlite> .headers ON
sqlite> .separator |
sqlite> .mode box
sqlite> SELECT
...> DISTINCT timezone AS time,
...> COUNT() AS count
...> FROM city
...> GROUP BY time
...> ORDER BY time DESC;

```

time	count
UTC+9	31
UTC+8	28
UTC+7	86
UTC+6	6
UTC+5	173
UTC+4	66
UTC+3	660
UTC+2	22
UTC+12	6
UTC+11	17
UTC+10	22

```

Run Time: real 0.003 user 0.001848 sys 0.000000
sqlite>

```

Рисунок 7. Задача 7

Выполнил индивидуальное задание: Нужно загрузить в SQLite датасет и сформировать более 5 запросов к таблицам БД. Все результаты выгрузить в формате CSV и JSON:

```
sqlite> .import --csv covid.csv covid
sqlite> .mode json
sqlite> .output 1.json
sqlite> .schema covid
sqlite> .output 2.json
sqlite> SELECT * FROM covid;
sqlite> .output 3.json
sqlite> SELECT * FROM covid limit 10;
sqlite> .output 4.json
sqlite> SELECT * FROM covid limit 15;
sqlite> .output 5.json
sqlite> SELECT * FROM covid limit 20;
sqlite> .mode csv
sqlite> .output 1.csv
sqlite> .schema covid
sqlite> .output 2.csv
sqlite> SELECT * FROM covid;
sqlite> .output 3.csv
sqlite> SELECT * FROM covid limit 10;
sqlite> .output 4.csv
sqlite> SELECT * FROM covid limit 15;
sqlite> .output 5.csv
sqlite> SELECT * FROM covid limit 20;
sqlite>
```

Рисунок 8. Индивидуальное задание

### Ответы на вопросы

1. Реляционные базы данных — базы данных, предназначенных для управления данными в структурированном виде (с использованием ключей, таблиц и т. д.)

СУБД — системы, которые обеспечивают доступ к данным и выполняет запросы пользователя и аналитику

2. Язык SQL предназначен для работы с данными в базах данных. С помощью него создаются, редактируются и управляются базы данные

3. Состав языка SQL мало чем отличается от состава других языков программирования: в нем имеются свои операторы, предназначенные для

управления таблицами баз данных, операторы для вставки и удаления данных и операторы управления правами пользования данными

4. Главная особенность СУБД SQLite — это то, что она представляет собой независимую от отдельного сервера встроенную базу данных. То есть, для СУБД SQLite работа происходит непосредственно с файлами на устройстве

5. Для Windows установка происходит с помощью исполняемого .exe файла, в то время как для Linux для установки необходим пакетный менеджер (apt, pacman и т. д.). Однако, на официальном сайте SQLite для Linux можно скачать архив с уже готовыми исполняемыми файлами, не требующими установки

6. Для создания базы данных SQLite достаточно ввести в командную строку команду `sqlite3 имябд.db`

7. Для определения текущей БД можно воспользоваться командой `SELECT имябд()`

8. Для создания и удаления таблиц в SQLite используются операторы `CREATE TABLE` и `DROP TABLE` соответственно

9. Первичный ключ — уникальный идентификатор каждой записи в таблице, позволяющий идентифицировать ее среди других записей

10. Для того, чтоб сделать ключ автоинкрементным, в поле `CREATE TABLE` можно добавить ключевое слово `AUTOINCREMENT`. Его особенность — автоматическое повышение значения при новых записях в таблицу

11. Инструкции `NOT` и `DEFAULT` используются в SQLite при создании таблицы. Первая инструкция указывает, что в таблице не может содержаться значение `NULL`, а вторая позволяет вводить любые значения

12. Внешние ключи в таблицах нужны для уставки связи между данными в таблице — как в пределах одной, так и между разными таблицами

13. Для вставки строки в таблицу БД можно использовать оператор `INSERT INTO`

14. Для выбора данных в таблице БД можно использовать команду SELECT

15. Для ограничения выборки данных можно использовать условие WHERE:

```
SELECT столб1, столб2, столб3, ...
```

```
FROM табл1
```

```
WHERE условие;
```

16. Для упорядочивания данных используется оператора ORDER BY

17. Для обновления записей в таблице БД можно использовать оператор UPDATE

18. Чтобы удалить записи из таблицы БД можно использовать оператор DELETE FROM

19. Для группировки данных в таблице БД можно использовать оператор GROUP BY

20. Для получения агрегатной функции в выборке из таблицы БД используются такие функции, как COUNT, MAX, MIN, SUM и AVG

21. Для того, чтобы объединить несколько таблиц, в операторе SELECT можно использовать оператор JOIN

22. Подзапросы в SQLite — запрос, включенный в другой запрос. Их можно использовать для выполнения более сложных операций выборки данных или редактирования таблицы

23. VIEW представляет собой виртуальные таблицы, с помощью которых можно в разы упростить работу со сложными запросами и для скрытия деталей структуры БД от пользователей

24. Существует несколько средств импорта данных в SQLite. Среди них выделяется встроенная команда SQLite .import, а также сторонние инструменты для работы с БД SQLite (например, DB Browser for SQLite)

25. .schema — команда, предназначенная для отображения схемы БД SQLite, в том числе данных о таблицах, индексах и представлениях

26. Для экспорта данных БД SQLite сначала нужно переключиться в соответствующий режим (.mode json или .mode csv) и воспользоваться запросом .output файл.json или .output файл.csv соответственно

27. Помимо CSV и JSON данные можно экспортировать в форматы таблиц Excel, в XML, SQL-скрипты и т. д.

Вывод: В ходе данной лабораторной работы были приобретены базовые навыки управления базами данными SQLite