

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»**

Выполнил:

Пустяков Андрей Сергеевич

2 курс, группа ИВТ-б-о-22-1,

09.03.01 «Информатика и  
вычислительная техника (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных  
систем», очная форма обучения

---

(подпись)

Руководитель практики:

Воронкин Р. А. кандидат технических  
наук, доцент, доцент кафедры  
инфокоммуникаций

---

(подпись)

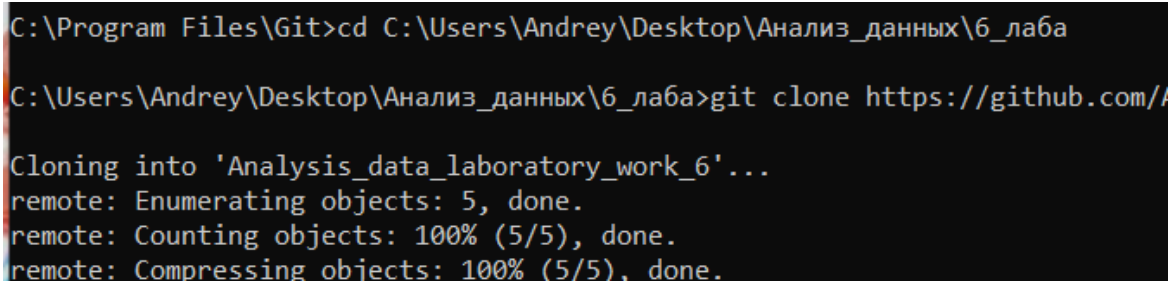
Ставрополь, 2024 г.

Тема: Основы работы с SQLite3.

Цель: исследовать базовые возможности системы управления базами данных SQLite3.

Ход работы:

Создание общедоступного репозитория на «GitHub», клонирование репозитория, редактирование файла «.gitignore (рис. 1).

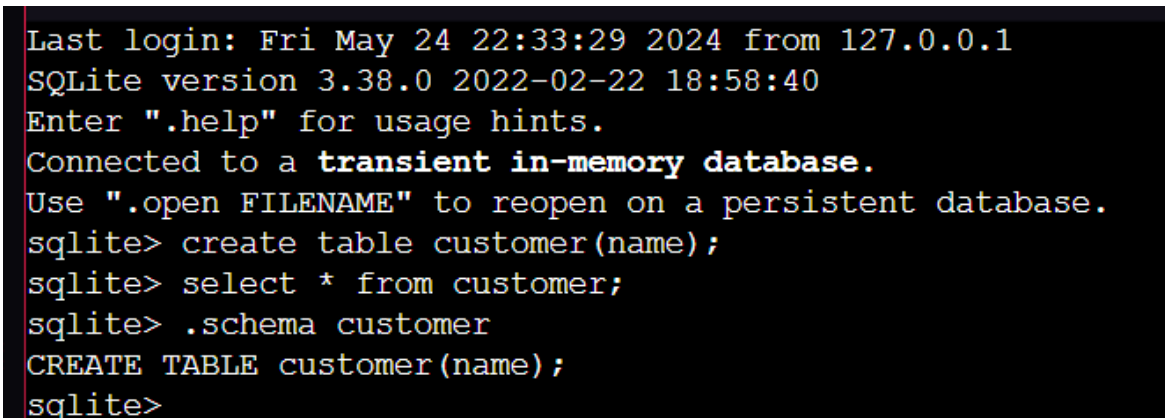


```
C:\Program Files\Git>cd C:\Users\Andrey\Desktop\Анализ_данных\6_лаба
C:\Users\Andrey\Desktop\Анализ_данных\6_лаба>git clone https://github.com/
Cloning into 'Analysis_data_laboratory_work_6'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
```

Рисунок 1 – Создание репозитория

Задание 7.

Необходимо выполнить некоторые команды в песочнице (создать таблицу с одним столбцом, вывести содержимое таблицы, вывести структуру таблиц) (рис. 2).



```
Last login: Fri May 24 22:33:29 2024 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> create table customer(name);
sqlite> select * from customer;
sqlite> .schema customer
CREATE TABLE customer(name);
sqlite>
```

Рисунок 2 – Команду в песочнице SQLite

Команда «.schema» вывела структуру всех таблиц в базе (CREATE TABLE customer(name);).

#### Задание 8.

Необходимо с помощью команды «.help» найти в песочнице команду, которая отвечает за вывод времени выполнения запроса (команда «.timer on») (рис. 3).

```
sqlite> .help t
.tables ?TABLE?      List names of tables matching LIKE pattern TABLE
.testcase NAME       Begin redirecting output to 'testcase-out.txt'
.testctrl CMD ...    Run various sqlite3_test_control() operations
.timeout MS          Try opening locked tables for MS milliseconds
.timer on|off        Turn SQL timer on or off
.trace ?OPTIONS?     Output each SQL statement as it is run
sqlite> select count(*);
1
sqlite> .timer on
sqlite> select count(*);
1
Run Time: real 0.000 user 0.000075 sys 0.000049
sqlite>
```

Рисунок 3 – Поиск команды вывода времени запроса

#### Задание 9.

Необходимо загрузить в песочницу файл формата «csv» и выполнить запрос (запрос вернул число 25) (рис. 4).

```
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
Run Time: real 0.001 user 0.000785 sys 0.000294
sqlite> █
```

Рисунок 4 – Загрузка файла и запрос

#### Задание 10.

Необходимо подгрузить тот же файл, но способом, доступным для старых версий СУБД (рис. 5).

```
sqlite> .mode csv
sqlite> .import city.csv city
sqlite> █
```

Рисунок 5 – Загрузка файла «city.csv»

#### Задание 11.

Необходимо написать в песочнице запрос, который посчитает количество городов для каждого часового пояса в сибирском и поволжском федеральных округах, вывести столбцы «timezone» и «city\_count», отсортировать по значению часового пояса (рис. 6).

```
sqlite> .header on
sqlite> SELECT Timezone, COUNT(*) AS City_Count FROM city WHERE Federal_district IN ("Сибирский", "Поволжский");
timezone, City_Count
UTC+7, 314
sqlite> .mode box
sqlite> SELECT Timezone, COUNT(*) AS City_Count FROM city WHERE Federal_district IN ("Сибирский", "Поволжский");
```

timezone	City_Count
UTC+7	314

```
sqlite> SELECT Timezone, COUNT(*) AS City_Count FROM city WHERE Federal_district IN ("Сибирский", "Поволжский") GROUP by Timezone ORDER by Timezone;
```

timezone	City_Count
UTC+3	101
UTC+4	41
UTC+5	58
UTC+6	6
UTC+7	86
UTC+8	22

Рисунок 6 – Вывод информации о количестве городов

#### Задание 12.

Необходимо написать в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару, применив при этом формулу для расчетов расстояний по координатам (рис. 7).

```
sqlite> WITH Samara AS (SELECT geo_lat, geo_lon FROM city WHERE city="Самара") SELECT GROUP_CONCAT(city, " , ") AS near FROM (SELECT city FROM city, Samara WHERE city.c
ity <> "Самара" ORDER BY (POWER(Samara.geo_lat - city.geo_lat, 2) + POWER(Samara.geo_lon - city.geo_lon, 2)) LIMIT 3);
```

near
Новокуйбышевск , Чапаевск , Кинель

Рисунок 7 – Вывод информации о трех ближайших городах

### Задание 13.

Необходимо написать в песочнице запрос, который посчитает количество городов в каждом часовом поясе и отсортировать запрос по количеству городов по убыванию (рис. 8). Выполнить такой же запрос, но в формате «csv».

```
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> .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 8 – Вывод информации в разных форматах

Выполнение индивидуальных заданий:

### Задание 1.

Необходимо загрузить свой датасет в формате «csv» в песочницу SQLite и сформировать 5 запросов, каждый из которых сохранить в формате «sql», «csv» и «json». В качестве датасета был использован датасет цен на видеокарты «NVIDIA» с 2000 года по 2024.

```
C:\Users\Andrey\Desktop\Анализ_данных\6_лаба\Analysis_data_laboratory_work_6\dataset>sqlite3 nvidia_database.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .mode csv
sqlite> .import NVIDIA.csv nvidia_table
sqlite>
```

Рисунок 9 – Загрузка данных из «csv»

```
sqlite> SELECT * FROM nvidia_table;
sqlite> .output stdout
sqlite> .output one.csv
sqlite> SELECT * FROM nvidia_table;
sqlite> .output one.csv
sqlite> .output stdout
sqlite> .mode json
sqlite> .output one.json
sqlite> SELECT * FROM nvidia_table;
sqlite> .output stdout
sqlite> .mode csv
sqlite> .output one.csv
sqlite> SELECT * FROM nvidia_table;
sqlite> .output stdout
sqlite> .output two.csv
```

Рисунок 10 – Сохранение запроса на разные типы файлов

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.

2. 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.

3. 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).

#### 4. 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.

#### 5. 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).

#### 6. Как создать базу данных SQLite?

В командной строке SQLite: `sqlite3 имя_базы_данных.db`.

Внутри SQLite: `CREATE DATABASE имя_базы_данных;`

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

В командной строке SQLite: `.database` или `.dbinfo`.

Внутри SQLite: `PRAGMA database_list;`

#### 8. Как создать и удалить таблицу в SQLite?

Создание таблицы:

```
CREATE TABLE название (  
поле1 тип1,  
... );
```

Удаление таблицы:

DROP TABLE название;

9. Что является первичным ключом в таблице?

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

10. Как сделать первичный ключ таблицы автоинкрементным?

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

11. 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.

12. 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.

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

Используйте оператор INSERT INTO.

14. Как выбрать данные из таблицы SQLite?

Используйте оператор SELECT.

15. Как ограничить выборку данных с помощью условия WHERE?

Используйте WHERE для установки условий:

SELECT column1, column2, ...

FROM table\_name

WHERE condition;



16. Как упорядочить выбранные данные?

Используйте ORDER BY.

17. Как выполнить обновление записей в таблице SQLite?

Используйте UPDATE.

18. Как удалить записи из таблицы SQLite?

Используйте DELETE.

19. Как сгруппировать данные из выборке из таблицы SQLite?

Используйте GROUP BY.

20. 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.:

21. Как выполнить объединение нескольких таблиц в операторе SELECT?

Используйте JOIN.

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

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

23. Каково назначение представлений VIEW в SQLite?

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

24. 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.

25. Каково назначение команды .schema ?

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

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

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

27. 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.

28. 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.

29. 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: в ходе выполнения лабораторной работы были исследованы базовые возможности системы управления базами данных SQLite3.