

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:  
Середа Кирилл Витальевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

Тема: Основы работы с SQLite3

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

Ход работы:

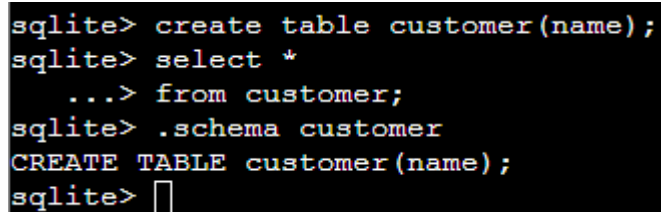
Задача 1 (рис.1):

create table customer(name);

select \*

from customer;

.schema customer

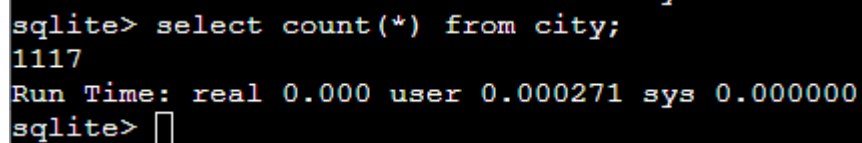


```
sqlite> create table customer(name);
sqlite> select *
...> from customer;
sqlite> .schema customer
CREATE TABLE customer(name);
sqlite> 
```

Рисунок 1 – Результат выполнения первой задачи

Задача 2:

Решите задачу: с помощью команды .help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса. Если ее включить, в результатах запроса добавится строка.

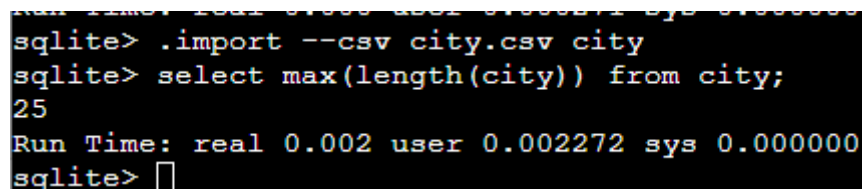


```
sqlite> select count(*) from city;
1117
Run Time: real 0.000 user 0.000271 sys 0.000000
sqlite> 
```

Рисунок 2 – Результат выполнения второй задачи

Задача 3:

Решите задачу: загрузите файл city.csv в песочнице (рис. 3)



```
Run Time: real 0.000 user 0.000271 sys 0.000000
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
Run Time: real 0.002 user 0.002272 sys 0.000000
sqlite> 
```

Рисунок 3 – Результат выполнения третьей задачи

Задача 4:

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. (рис. 4)

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

Рисунок 4 – Результат выполнения четвертой задачи

Задача 5:

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 (рис.5)

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

timee	count_city
timezone	1
UTC+9	62
UTC+8	56
UTC+7	172
UTC+6	12
UTC+5	346
UTC+4	132
UTC+3	1320
UTC+2	44
UTC+12	12
UTC+11	34
UTC+10	44

```
Run Time: real 0.003 user 0.003355 sys 0.000000
sqlite> 
```

Рисунок 5 – Результат выполнения пятой задачи

Задача 6: решите задачу: напишите в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару. (рис.6)

```

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

```

city	euclidean_distance
Новокуйбышевск	12.8298063265514
Чапаевск	24.7389598092284
Кинель	36.4840951529617

Рисунок 6 – Результат выполнения шестой задачи

Задача 7:

Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. (рис.7)

```

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

```

timee	count_city
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

sqlite> █

Рисунок 7 - Результат выполнения седьмой задачи

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

```
sqlite> .output z1.json
sqlite> .schema blood
sqlite> .mode csv
sqlite> .output z1.csv
sqlite> .schema blood
sqlite> .output z2.json
sqlite> .mode json
sqlite> SELECT * FROM blood;
sqlite> .mode csv
sqlite> .output z2.csv
sqlite> SELECT * FROM blood;
sqlite> .output z3.csv
sqlite> SELECT * FROM blood limit 4;
sqlite> .output z4.csv
sqlite> SELECT * FROM blood limit 10;
sqlite> .output z5.csv
sqlite> SELECT * FROM blood limit 15;
sqlite> .mode json
sqlite> .output z3.json
sqlite> SELECT * FROM blood limit 4;
sqlite> .output z4.json
sqlite> SELECT * FROM blood limit 10;
sqlite> .output z5.json
sqlite> SELECT * FROM blood limit 15;
sqlite>
```

Рисунок 8 – Индивидуальное задание

Вывод: в ходе практической работы были исследованы базовые возможности системы управления базами данных SQLite3