

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 29

Выполнил:
Саенко Андрей Максимович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. технических
наук, 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:

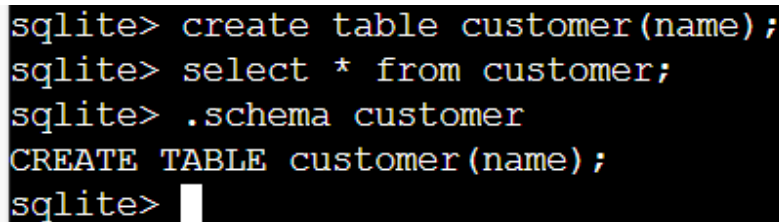
Выполнены задания:

Задание 7. Решите задачу: выполните в песочнице команды:

create table customer(name);

select * from customer;

.schema customer

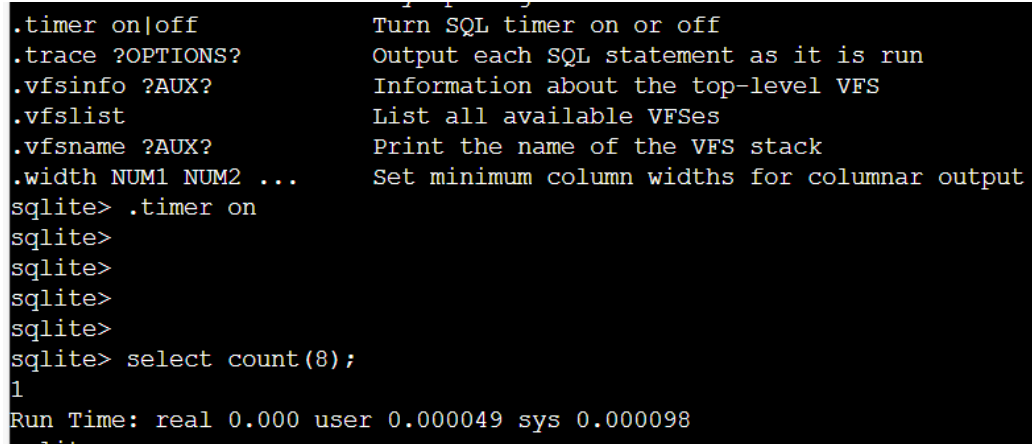


```
sqlite> create table customer(name);
sqlite> select * from customer;
sqlite> .schema customer
CREATE TABLE customer(name);
sqlite> 
```

Рисунок 1 – Выполнено задание 7

Была выведена структура таблицы.

Задание 8. Решите задачу: с помощью команды .help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса.



```
.timer on|off          Turn SQL timer on or off
.trace ?OPTIONS?      Output each SQL statement as it is run
.vfsinfo ?AUX?        Information about the top-level VFS
.vfslist              List all available VFSes
.vfsname ?AUX?        Print the name of the VFS stack
.width NUM1 NUM2 ...  Set minimum column widths for columnar output
sqlite> .timer on
sqlite>
sqlite>
sqlite>
sqlite> select count(8);
1
Run Time: real 0.000 user 0.000049 sys 0.000098
sqlite> 
```

Рисунок 2 – Выполнено задание 8 (команда - .timer on)

Задание 9. Решите задачу: загрузите файл city.csv в песочнице:

.import --csv city.csv city

Затем выполните такой запрос:

select max(length(city)) from city;

Какое число он вернул?

```
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
sqlite> 
```

Рисунок 3 – Выполнено задание 9 (запрос вернул число 25)

Задание 10. Решите задачу: загрузите файл city.csv в песочнице с помощью команды .import , но без использования опции --csv .

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

Рисунок 4 – Выполнено задание 10

Задание 11. 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> .open city-1.db
sqlite> .header on
sqlite> .mode box
sqlite> SELECT Timezone, count(*) AS City_Count
...> FROM city
...> WHERE Federal_district
...> IN ("Сибирский", "Приволжский")
...> 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 5 – Выполнено задание 11

Значение city_count для timezone = UTC+5 - 58.

Задание 12. Решите задачу: напишите в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару. Укажите в ответе названия этих трех городов через запятую в порядке удаления от Самары.

```
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.city <> "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 6 – Выполнено задание 12 (Ответ: Новокуйбышевск, Чапаевск, Кинель)

Задание 13. Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию. Выполните этот же запрос, но так, чтобы результат был в формате CSV, с заголовками, с разделителем «pipe» |.

```
sqlite> .header on
sqlite> .mode box
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> .header 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

Рисунок 7 – Выполнено задание 13

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.

```
sqlite> SELECT Education_Level, Customer_Age FROM test ORDER BY Customer_Age DESC;  
Education_Level,Customer_Age  
Graduate,68  
Doctorate,66  
"High School",66  
Unknown,65  
Graduate,64  
Unknown,63  
College,63  
Graduate,62  
Unknown,62  
"High School",61  
"High School",59  
Doctorate,59  
Graduate,59  
Graduate,59  
College,59  
Graduate,58  
Graduate,58  
Graduate,57  
Graduate,57  
College,57  
Unknown,57  
Graduate,57
```

Рисунок 8 – Выполнение индивидуального задания

Было выполнено пять запросов, результат вывода каждого из которых помещался в файлы csv и json.

Запросы:

1. SELECT * FROM dataset LIMIT 5;
2. SELECT Education_Level FROM test LIMIT 5;
3. SELECT Education_Level, Customer_Age FROM test ORDER BY Customer_Age DESC LIMIT 5;
4. SELECT count(*) FROM test LIMIT 5;
5. SELECT max(Customer_Age) FROM test;

Выводы: в ходе выполнения работы получены навыки написания SQL-запросов.