

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Отчет по лабораторной работе № 2.20

Основы работы с SQLite3

Выполнил студент группы ИВТ-б-о-20-1

Ищенко М.А.

Работа защищена « » _____ 20__ г.

Проверил(а) _____

Ставрополь 2022

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

Создан общедоступный репозиторий на GitHub

Выполнены в песочнице следующие команды, рис. 1

```
Last login: Sat Mar 19 19:30:41 2022 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 1

Использована команда, которая отвечает за вывод времени выполнения запроса. Если ее включить, в результатах запроса добавится строка, показывающая время выполнения запроса, рис. 2

```
sqlite> .help .timer
.timer on|off          Turn SQL timer on or off
sqlite> .timer on
sqlite> select count(*) from city;

+-----+
| count(*) |
+-----+
| 1117     |
+-----+

Run Time: real 0.000 user 0.000000 sys 0.000317
sqlite>
```

Рисунок 2

Выполнены следующие команды, в результате было возвращено число 25, рис. 1

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

Рисунок 3

Загружен файл city.csv в песочнице с помощью команды import, но без использования опции –csv, рис. 4

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

Рисунок 4

Написан запрос, считающий количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах, рис. 5

```
sqlite> select timezone,
...> count(*) city_count
...> from city
...> where federal_district
...> in ('Сибирский', 'Приволжский')
...> group by 1
...> order by 1 asc;
```

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

Рисунок 5

Написан в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару, рис. 6

```
sqlite> with samara as (
...> select geo_lat as lat,
...> geo_lon as lon
...> from city
...> where city = "Самара"
...> )
...> select city,
...> sqrt(power((lat - geo_lat), 2) + power((lon - geo_lon), 2))
...> as dist
...> from (samara, city)
...> Where city != 'Самара'
...> order by dist asc
...> limit 3;
```

city	dist
Новокуйбышевск	0.18569700863441
Чапаевск	0.358068603404667
Кинель	0.528066220190501

Рисунок 6

Написан в песочнице запрос, который посчитает количество городов в каждом часовом поясе, отсортировано по количеству городов по убыванию с разными форматами, рис. 7-9

```
sqlite> select timezone,  
...> count(*) city_count  
...> from city  
...> group by 1  
...> order by 2 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 – Формат box

```
sqlite> .mode csv  
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 8 – Формат csv

```
sqlite> .separator |  
sqlite> .header on  
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 9 – С заголовками и разделителем “|”

Индивидуальное задание

Загрузив в SQLite выбранный датасет. Сформируем более пяти запросов к таблицам БД. Выгрузим результат выполнения запросов в форматы CSV и JSON

Приведём примеры запросов.

Поиск количества фильмов, где доход был выше бюджета, рис. 10

```
sqlite> select count (*) as count
...> from movies
...> where revenue > budget;
count
-----
2585
sqlite>
```

Рисунок 10

Поиск первых 100 фильмов с наивысшим рейтингом, рис. 11

```
sqlite> select title, vote_average
...> from movies
...> where vote_count > 1000
...> order by vote_average desc
...> limit 100;
title                                     vote_average
-----
The Shawshank Redemption                 8.5
The Godfather                           8.4
Fight Club                             8.3
Schindler's List                        8.3
Spirited Away                          8.3
The Godfather: Part II                  8.3
Pulp Fiction                          8.3
Whiplash                               8.3
The Dark Knight                        8.2
The Green Mile                         8.2
Forrest Gump                          8.2
Once Upon a Time in America            8.2
GoodFellas                             8.2
Howl's Moving Castle                   8.2
The Empire Strikes Back                8.2
Psycho                                 8.2
```

Рисунок 11

Сортировка фильмов по тому, во сколько раз доходность превысила бюджет, рис. 12

```
sqlite> select revenue / budget as profitability,  
...> title, release_date  
...> from movies  
...> where budget > 100000  
...> order by profitability desc;  
profitability  title  release_date  
-----  
420           Open Water  2004-08-06  
311           Bambi  1942-08-14  
263           Night of the Living Dead  1968-10-01  
250           Mad Max  1979-04-12  
233           Halloween  1978-10-25  
180           American Graffiti  1973-08-01  
129           Once  2007-03-23  
124           Snow White and the Seven Dwarfs  1937-12-20  
117           Rocky  1976-11-21
```

Рисунок 12

Поиск самых популярных фильмов с 2000 года, рис. 13

```
sqlite> select title, popularity, release_date  
...> from movies  
...> where release_date > '2000-01-01'  
...> order by popularity desc;  
title  popularity  release_date  
-----  
Minions  875  2015-06-17  
Interstellar  724  2014-11-05  
Deadpool  514  2016-02-09  
Guardians of the Galaxy  481  2014-07-30  
Mad Max: Fury Road  434  2015-05-13  
Jurassic World  418  2015-06-09  
Pirates of the Caribbean: The Curse of the Black Pearl  271  2003-07-09  
Dawn of the Planet of the Apes  243  2014-06-26  
The Hunger Games: Mockingjay - Part 1  206  2014-11-18  
Big Hero 6  203  2014-10-24  
Terminator Genisys  202  2015-06-23  
Captain America: Civil War  198  2016-04-27  
Whiplash  192  2014-10-10  
The Dark Knight  187  2008-07-16  
Inception  167  2010-07-14  
The Martian  167  2015-09-30  
Frozen  165  2013-11-27  
Batman v Superman: Dawn of Justice  155  2016-03-23
```

Рисунок 13

Поиск режиссеров с наибольшим количеством фильмов, рис. 14

```
sqlite> select name
...> from directors
...> join movies on directors.id = movies.director_id
...> group by director_id
...> order by count(name) desc
...> limit 10;
name
-----
Steven Spielberg
Woody Allen
Clint Eastwood
Martin Scorsese
Spike Lee
Robert Rodriguez
Ridley Scott
Renny Harlin
Steven Soderbergh
Oliver Stone
```

Рисунок 14

Поиск фильмов, режиссёрам которых является Вуди Аллен, рис. 15

```
sqlite> select original_title from movies
...> join directors on directors.id = movies.director_id
...> where directors.name = 'Woody Allen';
original_title
-----
Midnight in Paris

The Curse of the Jade Scorpion

To Rome with Love

Bullets Over Broadway

Deconstructing Harry

Everyone Says I Love You

Blue Jasmine
```

Рисунок 15

Вывод: в ходе занятия были исследованы базовые возможности системы управления базами данных SQLite3.