

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и телекоммуникаций

Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
Основы SQLite

Выполнил:

Ярыгин Альберт Андреевич

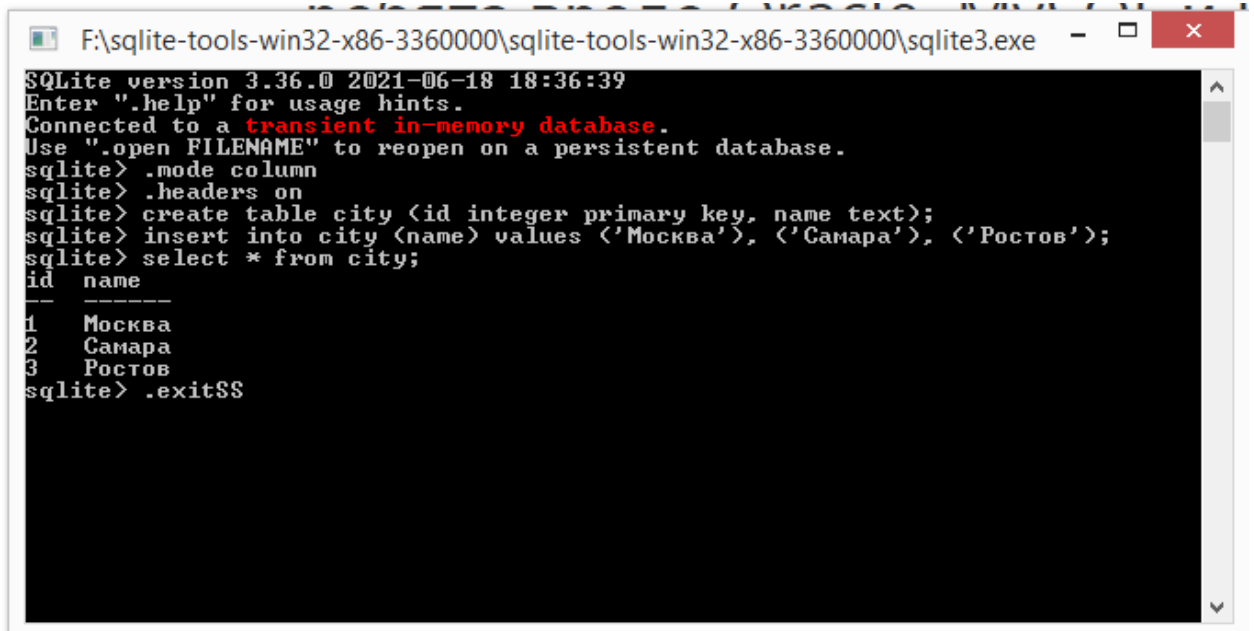
3 курс, группа ИВТ-б-о-19-1

Ставрополь, 2021 г

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

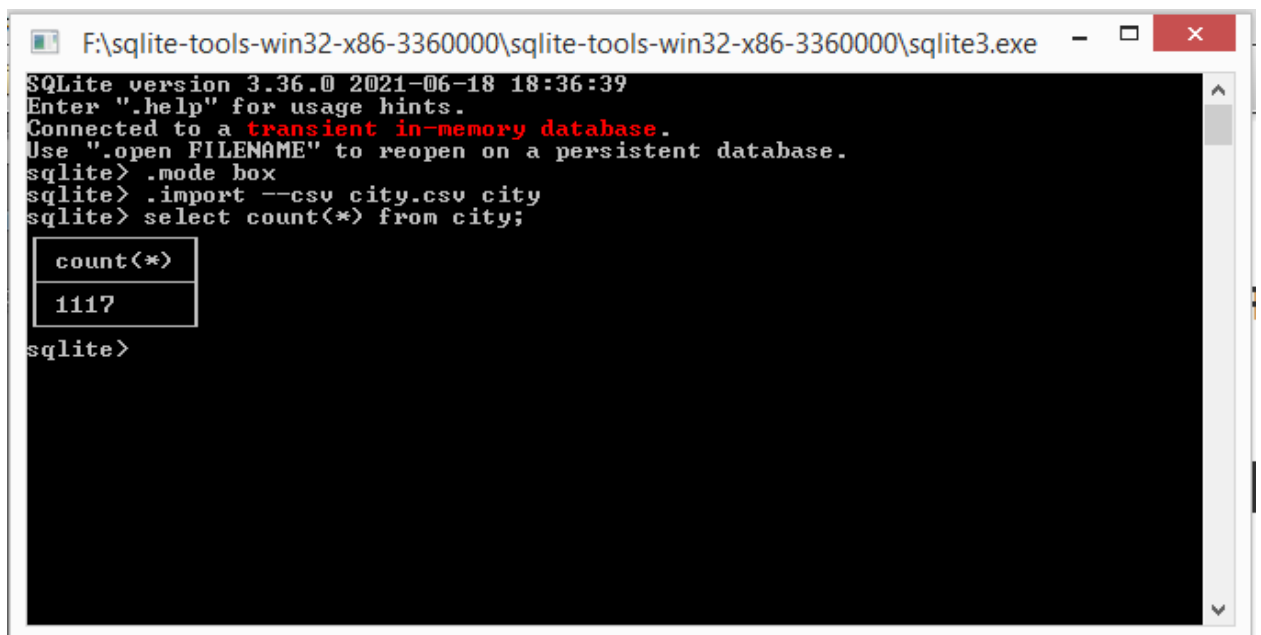
Ход работы:

- 1) Создание таблицы из data seta city:



```
F:\sqlite-tools-win32-x86-3360000\sqlite-tools-win32-x86-3360000\sqlite3.exe
SQLite version 3.36.0 2021-06-18 18:36:39
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .mode column
sqlite> .headers on
sqlite> create table city (id integer primary key, name text);
sqlite> insert into city (name) values ('Москва'), ('Самара'), ('Ростов');
sqlite> select * from city;
id  name
---  ---
1   Москва
2   Самара
3   Ростов
sqlite> .exit$$
```

- 2) Скачали файл city.csv, и загрузили в него данные:



```
F:\sqlite-tools-win32-x86-3360000\sqlite-tools-win32-x86-3360000\sqlite3.exe
SQLite version 3.36.0 2021-06-18 18:36:39
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .mode box
sqlite> .import --csv city.csv city
sqlite> select count(*) from city;
count(*)
1117
sqlite>
```

- 3) Создали таблицу city и загрузили в него все столбцы из файла city.csv

```
F:\sqlite-tools-win32-x86-3360000\sqlite-tools-win32-x86-3360000\sqlite3.exe
"oktmo" TEXT,
"tax_office" TEXT,
"timezone" TEXT,
"geo_lat" TEXT,
"geo_lon" TEXT,
"population" TEXT,
"foundation_year" TEXT
);
sqlite> select federal_district, city, population
...> from city limit 10;
```

federal_district	city	population
Южный	Адыгейск	12689
Южный	Майкоп	144055
Сибирский	Горно-Алтайск	62861
Сибирский	Алейск	28528
Сибирский	Барнаул	635585
Сибирский	Белокуриха	15072
Сибирский	Бийск	203826
Сибирский	Горняк	13040
Сибирский	Заринск	47035
Сибирский	Змеиногорск	10569

```
sqlite>
```

4) Вывели количество городов из каждого федерального округа:

```
F:\sqlite-tools-win32-x86-3360000\sqlite-tools-win32-x86-3360000\sqlite3.exe
```

Сибирский	Бийск	203826
Сибирский	Горняк	13040
Сибирский	Заринск	47035
Сибирский	Змеиногорск	10569

```
sqlite> select
...> federal_district as district,
...> count(*) as city_count
...> from city
...> group by 1
...> order by 2 desc
...> ;
```

district	city_count
Центральный	304
Приволжский	200
Северо-Западный	148
Уральский	115
Сибирский	114
Южный	96
Дальневосточный	82
Северо-Кавказский	58

```
sqlite>
```

5) Отфильтровали таблицу по запросам:

```
sqlite> select federal_district, city, population
...> from city limit 10;
```

federal_district	city	population
Южный	Адыгейск	12689
Южный	Майкоп	144055
Сибирский	Горно-Алтайск	62861
Сибирский	Алейск	28528
Сибирский	Барнаул	635585
Сибирский	Белокуриха	15072
Сибирский	Бийск	203826
Сибирский	Горняк	13040
Сибирский	Заринск	47035
Сибирский	Змеиногорск	10569

```
sqlite> select
...> federal_district as district,
...> count(*) as city_count
...> from city
...> group by 1
...> order by 2 desc
...> ;
```

district	city_count
Центральный	304
Приволжский	200
Северо-Западный	148
Уральский	115
Сибирский	114
Южный	96
Дальневосточный	82
Северо-Кавказский	58

```
sqlite> select address
...> from city
...> where city like '%Красный%';
```

address
Ростовская обл, г Красный Сулин
Саратовская обл, г Красный Кут
Тверская обл, г Красный Холм

```
sqlite> select region, city, foundation_year
...> from city
...> where foundation_year between 1990 and 2020;
```

region	city	foundation_year
Ингушетия	Магас	1995
Татарстан	Иннополис	2012

```
sqlite> select region, city, foundation_tear
...> from city
...> where
...> foundation_year between date('now', '-30 years') and date('now')
...> ;
```

Error: no such column: foundation_tear

```
sqlite> select region, city, foundation_year
...> from city
...> where
...> foundation_year between date('now', '-30 years') and date('now')
...> ;
```

region	city	foundation_year
Ингушетия	Магас	1995
Татарстан	Иннополис	2012

```
sqlite> select count(*)
...> from city
...> where
...> federal_district in ('Приволжский', 'Уральский')
...> ;
```

count(*)
315

```
sqlite>
```

6) Вывели количество городов основанных в каждом веке:

```
sqlite> with history as (  
...> select  
...> city,  
...> <foundation_year/100>+1 as century  
...> from city  
...> )  
...> select  
...> century || '-й век' as dates,  
...> count(*) as city_count  
...> from history  
...> group by century  
...> order by century desc  
...> ;
```

dates	city_count
21-й век	1
20-й век	263
19-й век	189
18-й век	191
17-й век	137
16-й век	79
15-й век	39
14-й век	38
13-й век	27
12-й век	44
11-й век	8
10-й век	6
9-й век	4
5-й век	2
3-й век	1
1-й век	88

7) Выгружаем данные из таблицы city в таблицу samara:

```
sqlite> .mode csv  
sqlite> select kladr_id, city  
...> from city  
...> where region = 'Самарская'  
...> limit 3;  
kladr_id,city  
630000002000000,"Жигулевск"  
630000010000000,"Кинель"  
63017001000000,"Нефтегорск"  
sqlite>
```

Выполнение заданий:

8) Создали таблицу с единственным столбцом name:

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

9) Добавили время выполнения запроса каждой команды:

```

sqlite> .timer on
sqlite> select count(*) from city;
Run Time: real 0.000 user 0.000000 sys 0.000000
Error: no such table: city
sqlite> .import --csv city.csv city
sqlite> .import --csv samara.csv samara
sqlite> select count(*) from city;
1117
Run Time: real 0.000 user 0.000000 sys 0.000000
sqlite> select max(length(city)) from city;
25
Run Time: real 0.000 user 0.000000 sys 0.000000

```

- 10) Посчитали количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах:

```

sqlite> with new_pirates as (
...> select timezone as zones from city where federal_district in ('Приволжский','Уральский'))
...> select zones, count(*) as city_count from new_pirates group by zones order by zones desc;
zones,city_count
UTC+5,173
UTC+4,41
UTC+3,101
sqlite> with new_pirates as (
...> select timezone as zones from city)
...> select zones, count(*) as city_count from new_pirates group by zones order by zones desc;
Error: no such table: new
sqlite> with new_pirates as (
...> select timezone as zones from city)
...> select zones, count(*) as city_count from new_pirates group by zones order by zones desc;
zones,city_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
sqlite> with new_pirates as (
...> select timezone as zones from city)
...> select zones, count(*) as city_count from new_pirates group by zones order by count(*) desc;
zones,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>

```

- 11) Нашли три ближайших к Самаре города:

```

sqlite> with target as (
...> select geo_lat, geo_lon
...> from city
...> where city = 'Самара'
...> )
...>
...> select city
...> from (
...> select city,
...> geo_lat,
...> geo_lon,
...> (select geo_lat from target) as target_geo_lat,
...> (select geo_lon from target) as target_geo_lon
...> from city
...> where city != 'Самара'
...> )
...> order by sqrt(
...> power(geo_lat - target_geo_lat, 2) + power(geo_lon - target_geo_lon,2)
...> )
...> limit 3;
Новокуйбышевск
Чапаевск
Кинель
sqlite>

```

- 12) Посчитали количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах и отсортировали их по убыванию:

```

sqlite> with wow as (
...> select timezone as zones from city)
...> select zones, count(*) as city_count from wow group by zones order by count(*) desc;

```

zones	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> .headers on
sqlite> with wow as (
...> select timezone as zones from city)
...> select zones, count(*) as city_count from wow group by zones order by count(*) desc;

```

zones	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> .headers on
sqlite> .separator |
sqlite> with wow as (
...> select timezone as zones from city)
...> select zones, count(*) as city_count from wow group by zones order by count(*) desc;
zones|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>

```

Индивидуальное задание:

```
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .mode csv
sqlite> .import city.csv city
sqlite> .once mikhailovsk.csv
sqlite> .headers on
sqlite> select kladr_id, population, timezone, foundation_year, city from city
where region = 'Ставропольский'
...> ;
kladr_id, population, timezone, foundation_year, city
2600600100000,32736,UTC+3,1782,"Благодарный"
2600700100000,64628,UTC+3,1799,"Буденновск"
2600000900000,72126,UTC+3,1777,"Георгиевск"
2600000200000,100969,UTC+3,1825,"Ессентуки"
2600000300000,24496,UTC+3,1810,"Железноводск"
2602300100000,35790,UTC+3,1781,"Зеленокумск"
2601000100000,40546,UTC+3,1895,"Изобильный"
2601100100000,26055,UTC+3,1860,"Ипатово"
2600000400000,128502,UTC+3,1803,"Кисловодск"
2600000500000,22540,UTC+3,1953,"Лермонтов"
2601700200000,76715,UTC+3,1878,"Минеральные Воды"
2602700100000,71018,UTC+3,1784,"Михайловск"
2600000600000,118351,UTC+3,1825,"Невинномысск"
2601800100000,27700,UTC+3,1958,"Нефтекумск"
2601900100000,26759,UTC+3,1804,"Новоалександровск"
2601200100000,26556,UTC+3,1777,"Новопавловск"
2600000700000,142397,UTC+3,1780,"Пятигорск"
2602100100000,38520,UTC+3,1786,"Светлоград"
2600000100000,398266,UTC+3,1777,"Ставрополь"
sqlite> .headers on
sqlite> .once mikhailovsk.csv
sqlite> select kladr_id, population, timezone, foundation_year, city from city
where region = 'Ставропольский'
...> ;
sqlite> select kladr_id, population, timezone, foundation_year, city from city
where region = 'Ставропольский'
...> ;
...> ;
kladr_id, population, timezone, foundation_year, city
2600600100000,32736,UTC+3,1782,"Благодарный"
2600700100000,64628,UTC+3,1799,"Буденновск"
2600000900000,72126,UTC+3,1777,"Георгиевск"
2600000200000,100969,UTC+3,1825,"Ессентуки"
2600000300000,24496,UTC+3,1810,"Железноводск"
2602300100000,35790,UTC+3,1781,"Зеленокумск"
2601000100000,40546,UTC+3,1895,"Изобильный"
2601100100000,26055,UTC+3,1860,"Ипатово"
2600000400000,128502,UTC+3,1803,"Кисловодск"
2600000500000,22540,UTC+3,1953,"Лермонтов"
2601700200000,76715,UTC+3,1878,"Минеральные Воды"
2602700100000,71018,UTC+3,1784,"Михайловск"
2600000600000,118351,UTC+3,1825,"Невинномысск"
2601800100000,27700,UTC+3,1958,"Нефтекумск"
2601900100000,26759,UTC+3,1804,"Новоалександровск"
2601200100000,26556,UTC+3,1777,"Новопавловск"
2600000700000,142397,UTC+3,1780,"Пятигорск"
2602100100000,38520,UTC+3,1786,"Светлоград"
2600000100000,398266,UTC+3,1777,"Ставрополь"
```

Вывод: В ходе выполнения лабораторной работы исследовали базовые возможности системы управления базами данных SQLite.

Контрольные вопросы:

1. Какие существуют средства для импорта данных в SQLite?

```
.import —csv city.csv city
```

```
.mode csv
```

```
.import city.csv city
```

2. В чем недостатки локальных и централизованных СКВ?

Локальные СКВ

Легко запутаться в файлах и в каком конкретно ты находишься, можно изменить не тот файл или сохранить в не нужной директории

Централизованные

Централизованный сервер является уязвимым местом всей системы. Если сервер выключается на час, то в течение часа разработчики не могут взаимодействовать, и никто не может сохранить новые версии. Если же повреждается диск с центральной базой данных и нет резервной копии, вы теряете абсолютно всё - всю историю проекта, разве что за исключением нескольких рабочих версий, сохранившихся на рабочих машинах пользователей. Локальные системы управления версиями подвержены той же проблеме: если вся история проекта хранится в одном месте, вы рискуете потерять всё.

3. Каково назначение команды .schema ?

показывает список и структуру всех таблиц в базе

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

Группировка производится, когда в оператора group используется функция by

5. Каково назначение "табличных выражений" в SQLite?

Наглядная демонстрация содержимого в таблице

6. Как осуществляется экспорт данных из SQLite в форматы CSV и JSON?

```
sqlite> .mode csv
```

```
sqlite> .once samara.csv
```

```
sqlite> select kladr_id, city from city where region = 'Самарская';
```

```
sqlite> .exit
```

```
.mode json
```

```
select kladr_id, city
```

```
from city where region = 'Самарская' limit 3;
```

7. Какие еще форматы для экспорта данных Вам известны?

```
.mode insert cities
```

```
.mode markdown
```

```
.mode html
```