

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт цифрового развития

Кафедра инфокоммуникаций.

Дисциплина: Базы данных

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

Основы SQLite

**Выполнил:** студент 3 курса  
09.03.01 «Информатика и вычислительная  
техника» группы ИВТ-б-о-19-1  
Демьяненко Александр Владимирович

**Проверил:**  
Воронкин Роман Александрович

Работа защищена с оценкой:

---

Ставрополь, 2021

# Лабораторная работа №1

Тема: Основы SQLite

Цель: исследовать базовые возможности системы управления базами данных SQLite.

Выполнение работы:

## 1.Создание таблицы базы данных с городами

```
sqlite> .mode box
sqlite> create table city (id integer primary key, name text);
sqlite> insert into city (name) values ('Москва'), ('Санкт-Петербург'), ('Новосибирск');
sqlite> select * from city;
```

id	name
1	Москва
2	Санкт-Петербург
3	Новосибирск

## 2.Отображение количества городов в city.csv

```
sqlite> .mode box
sqlite> .import --csv city.csv city
sqlite> select count(*) from city;
```

count(*)
1117

```
sqlite>
```

## 3.Просмотр столбцов, которые имеются в таблице

```

sqlite> .schema city
CREATE TABLE IF NOT EXISTS "city"(
  "address" TEXT,
  "postal_code" TEXT,
  "country" TEXT,
  "federal_district" TEXT,
  "region_type" TEXT,
  "region" TEXT,
  "area_type" TEXT,
  "area" TEXT,
  "city_type" TEXT,
  "city" TEXT,
  "settlement_type" TEXT,
  "settlement" TEXT,
  "kladr_id" TEXT,
  "fias_id" TEXT,
  "fias_level" TEXT,
  "capital_marker" TEXT,
  "okato" TEXT,
  "oktmo" TEXT,
  "tax_office" TEXT,
  "timezone" TEXT,
  "geo_lat" TEXT,
  "geo_lon" TEXT,
  "population" TEXT,
  "foundation_year" TEXT
);
sqlite>

```

#### 4. Просмотр содержимого

```

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>

```

#### 5. Просмотр количества городов в каждом из федеральных округов

```
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> _
```

6.Выполнение точно такого же запроса через federal\_district

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

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

```
sqlite>
```

7. Просмотр городов с употреблением слова Красный в названии.

```
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>
```

8.Просмотр городов, которые появились в промежутке с 199 по 2020

```
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>
```

9.Просмотр количества городов в Приволжском и Уральских округах

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

count(*)
315

```
sqlite> _
```

10.Сколько городов было основано 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

```
sqlite>
```

11. Вывод по умолчанию, без заголовка, разделителем является запятая

```

sqlite> .mode csv
sqlite> select kladr_id, city
...> from city
...> where region = 'Самарская'
...> limit 3;
5300000200000,"Жигулевск"
5300001000000,"Кинель"
5301700100000,"Нефтегорск"
sqlite>

```

12. Вывод по умолчанию, с заголовком

```

sqlite> .mode csv
sqlite> .headers on
sqlite> select kladr_id,city
...> from city
...> where region = 'Самарская'
...> limit 3;
kladr_id,city
6300000200000,"Жигулевск"
6300001000000,"Кинель"
6301700100000,"Нефтегорск"
sqlite>

```

### 13. Вывод с другим разделителем

```

sqlite> .mode csv
sqlite> .headers on
sqlite> .separator ;
sqlite> select kladr_id,city
...> from city
...> where region = 'Самарская'
...> limit 3;
kladr_id;city
6300000200000;"Жигулевск"
6300001000000;"Кинель"
6301700100000;"Нефтегорск"
sqlite>

```

### 14. Подключение JSON

```

sqlite> .mode json
sqlite> select kladr_id, city
...> from city
...> where region = 'Самарская'
...> limit 3;
[{"kladr_id":"6300000200000","city":"Жигулевск"},
{"kladr_id":"6300001000000","city":"Кинель"},
{"kladr_id":"6301700100000","city":"Нефтегорск"}]

```

### 15. Использование команды INSERT

```

sqlite> .mode insert cities
sqlite> select kladr_id, city
...> from city
...> where region = 'Самарская'
...> limit 3;
INSERT INTO cities VALUES('6300000200000','Жигулевск');
INSERT INTO cities VALUES('6300001000000','Кинель');
INSERT INTO cities VALUES('6301700100000','Нефтегорск');

```

### 16. Изучение возможностей и вывода markdown

```

sqlite> .mode markdown
sqlite> select kladr_id, city
...> from city
...> where region = 'Самарская'
...> limit 3;
| kladr_id | city |
|-----|-----|
| 6300000200000 | Жигулевск |
| 6300001000000 | Кинель |
| 6301700100000 | Нефтегорск |
sqlite> .mode html
sqlite> select kladr_id, city
...> from city
...> where region = 'Самарская'
...> limit 3;
<TR><TD>6300000200000</TD>
<TD>Жигулевск</TD>
</TR>
<TR><TD>6300001000000</TD>
<TD>Кинель</TD>
</TR>
<TR><TD>6301700100000</TD>
<TD>Нефтегорск</TD>
</TR>
sqlite>

```

17. Попробуем загрузить его в таблицу samara

```

sqlite> .import --csv samara.csv samara
sqlite> .mode box
sqlite> select * from samara limit 5;

```

6300000200000	Жигулевск
6300001000000	Кинель
6301700100000	Нефтегорск
6300000300000	Новокуйбышевск
6300000400000	Октябрьск
6300000500000	Отрадный

18. Подключение заголовков, разделителей

```

sqlite> create table samara (kladr_id, name);
sqlite> .mode csv
sqlite> .headers on
sqlite> .separator |
sqlite> .import samara.csv samara
sqlite> .mode box
sqlite> select * from samara limit 5;

```

kladr_id	name
6300000200000	Жигулевск
6300001000000	Кинель
6301700100000	Нефтегорск
6300000300000	Новокуйбышевск
6300000400000	Октябрьск

```

sqlite>

```



## 19.Выполнение команд

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

## 20.Загрузка команд

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

max(length(city))
25

```
sqlite>
```

## 21.Отображение таймера выполнения

```
sqlite> select count(*) from city;
count(*)
3353
Run Time: real 0.003 user 0.000000 sys 0.000000
```

## 22.Импорт без --csv

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

## 23.Выполнение 11 задания

```

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>

```

## 24.Выполнение 12 задания

```

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>

```

## 25. Выполнение 13 задания

```
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> _
```

## 26.Индивидуальное задание

```

sqlite> .once stavropol.csv
sqlite> select kladr_id,city from city where region = 'Ставропольский';
sqlite> .mode csv
sqlite> select kladr_id, city
...> from city
...> where region = 'Ставропольский'
...> limit 3;
kladr_id,city
2600600100000,"Благодарный"
2600700100000,"Буденновск"
2600000900000,"Георгиевск"
sqlite> .headers on
sqlite> select kladr_id, city
...> from city
...> where region = 'Ставропольский'
...> limit 3;
kladr_id,city
2600600100000,"Благодарный"
2600700100000,"Буденновск"
2600000900000,"Георгиевск"
sqlite> select address
...> from region = 'Ставропольский'
...> where region = 'Ставропольский' like '%Красный';
Error: near "=": syntax error
sqlite> .import --csv stavropol.csv stavropol
sqlite> select count(*) from stavropol;
count(*)
19
sqlite> .mode list
sqlite> .headers on
sqlite> .separator ,
sqlite> select kladr_id, city
...> from city
...> where region = 'Ставропольский'
...> limit 3;
kladr_id,city
2600600100000,Благодарный
2600700100000,Буденновск
2600000900000,Георгиевск
sqlite> .mode markdown
sqlite> select kladr_id, city
...> from city
...> where region = 'Ставропольский'
...> limit 3;


| kladr_id      | city        |
|---------------|-------------|
| 2600600100000 | Благодарный |
| 2600700100000 | Буденновск  |
| 2600000900000 | Георгиевск  |


sqlite> _

```

## Запуск (или открытие) программы

1. Выполните команду *Пуск>Все программы*. Найдите название программы в отобразившемся списке *Все программы* и щелкните на нем. Если щелкнуть на объекте с пиктограммой папки, отобразится список значков находящихся в этой папке программ. Затем

щелкните на значке программы в этом вложенном списке, и она откроется.

2. Дважды щелкните на иконке программы, расположенном на рабочем столе.

3. Щелкните на объекте, расположенном на панели задач. По умолчанию панель задач отображается на экране. Если же ее не видно, нажмите клавишу <Windows> на клавиатуре. Затем щелкните на значке программы на панели задач, находящейся справа от кнопки *Пуск*.

4. Если вы недавно использовали программу и сохранили документ, выберите ее из списка недавно использованных программ, отображаемых при первом открытии меню кнопки *Пуск*. Затем в открывшемся списке щелкните на документе, созданном в этой программе.

Вывод: в ходе выполнения лабораторной работе исследовал 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
```