

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

Кафедра инфокоммуникаций.

Дисциплина: Кроссплатформенное программирование

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Управление таблицами и данными

Выполнил: студент 3 курса
09.03.01 «Информатика и вычислительная
техника» группы ИВТ-б-о-19-1
Винников Никита Александрович

Проверил:
Воронкин Роман Александрович

Работа защищена с оценкой:

Лабораторная работа №1

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

Ход работы:

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

```
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .mode column
sqlite> .headers on
sqlite> .import --csv city.csv city
sqlite> select count(*) from city;
count(*)
-----
1117
```

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

```
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
```

3) Вывели количество городов из каждого федерального округа:

```
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
...> 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>
```

4) Отфильтровали таблицу по запросам:

```
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 date('now','-30 years') and date('now')
...> ;
region      city      foundation_year
-----
Ингушетия   Магас     1995
Татарстан   Иннополис 2012
```

5) 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
```

6) Выгружаем данные из таблицы city в таблицу samara:

```
sqlite> .once samara.csv
sqlite> .mode csv
sqlite> select kladr_id, city from city where region = 'Самарская';
kladr_id,city
63000000200000,"Жигулевск"
63000001000000,"Кинель"
63017001000000,"Нефтегорск"
63000000300000,"Новокуйбышевск"
63000000400000,"Октябрьск"
63000000500000,"Отрадный"
63000000900000,"Похвистнево"
63000000100000,"Самара"
63000000800000,"Сызрань"
63000000700000,"Тольятти"
63000000600000,"Чапаевск"
```

Выполнение заданий:

7) 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
```

8) 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
```

9) 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>
```

10) 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>
```

Индивидуальное задание:

Импортировал data set spotify

```
sqlite> .import --csv spotify.csv spotify
```

Сделал запросы и импортировали запросы в формат json, sql, csv.

```

sqlite> select
...> artist as artist
...> from spotify
...> group by 1;
sqlite> .once 2.csv
sqlite> select album
...> from spotify limit 10;
sqlite> .mode json
sqlite> once 1.json
...> select album
...> from spotify limit 10;
Error: near "once": syntax error
sqlite> .once 1.json
sqlite> select album
...> from spotify limit 10;
sqlite> once 2.json
...> select
...> artist as artist
...> from spotify
...> group by 1;
Error: near "once": syntax error
sqlite> .once 2.json
sqlite> select
...> artist as artist
...> from spotify
...> group by 1;
sqlite> .once 3.json
sqlite> select track_name
...> from spotify
...> where spotify like 'D%';
Error: no such column: spotify
sqlite> .once 3.json
sqlite> select track_name
...> from spotify
...> where track_name like 'D%';
sqlite> .once 3.csv
sqlite> select track_name
...> from spotify
...> where track_name like 'D%';
sqlite> .once 4.json
sqlite> select artist
...> from spotify
...> where artist = 'Justin Bieber';
sqlite> .once 4.csv
sqlite> select artist
...> from spotify
...> where artist = 'Justin Bieber';
sqlite> .once 5.json
sqlite> select genre
...> from spotify
...> where genre = 'Hip-Hop/Rap';
sqlite> .once 5.csv
sqlite> select genre
...> from spotify
...> where genre = 'Hip-Hop/Rap';

```

```

sqlite> select genre
...> from spotify
...> where genre = 'Hip-Hop/Rap';
sqlite> .once 5.csv
sqlite> select genre
...> from spotify
...> where genre = 'Hip-Hop/Rap';
sqlite> .mode markdown
sqlite> .once 1.sql
sqlite> select genre
...> from spotify
...> where genre = 'Hip-Hop/Rap';
sqlite> .mode markdown
sqlite> .once 2.sql
sqlite> select artist
...> from spotify
...> where artist = 'Justin Bieber';
sqlite> .mode markdown
sqlite> .once 3.sql
sqlite> select track_name
...> from spotify
...> where track_name like 'D%';
sqlite> .mode markdown
sqlite> .once 4.sql
sqlite> select
...> artist as artist
...> from spotify
...> group by 1;
sqlite> .mode markdown
sqlite> .once 5.sql
sqlite> select album
...> from spotify
...> limit 10;
sqlite>

```

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
```