

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 15

Основы работы с SQLite3

По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-20-1

Бобров Н. В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2022

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

Ход работы:

1. Создал общедоступный репозиторий и клонировал его на локальный сервер.
2. Выполнил задания по пункту 7-13.
- 2.7. Выполнить команды в песочнице.

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

Рисунок 1 – Задание 1

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

```
sqlite> .timer on  
sqlite> select count(*) from city;  


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

  
Run Time: real 0.000 user 0.000088 sys 0.000032
```

Рисунок 2 – Задание 2

- 2.9. Решите задачу: загрузите файл city.csv и выполнить запрос. Какое число он вернул?

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


| max(length(city)) |
|-------------------|
| 25                |

  
Run Time: real 0.003 user 0.000439 sys 0.000185
```

Рисунок 3 – Задание 3

- 2.10. 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
```

Рисунок 4 – Задание 4

2.11. 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> select
...> timezone, count(city) as city_count
...> from city where federal_district = 'Приволжский' or federal_district =
'Sибирский'
...> group by timezone
...> order by timezone ASC
...> ;
timezone,city_count
UTC+3,303
UTC+4,123
UTC+5,174
UTC+6,18
UTC+7,258
UTC+8,66
Run Time: real 0.004 user 0.001503 sys 0.000000
sqlite> █
```

Рисунок 5 – Задание 5

2.12. Решите задачу: напишите в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару. Если не получится, не расстраивайтесь — задача действительно непростая. Вернитесь к ней, когда пройдете все модули курса — и увидите, как все изменилось. Укажите в ответе названия этих трех городов через запятую в порядке удаления от Самары.

```
sqlite> with geo_las as (select geo_lat as geo_las from city where city = 'Самара'),
...> geo_los as (select geo_lon as geo_los from city where city = 'Самара'),
...> geo_lam as (select geo_lat as geo_lam, city from city), geo_lou as (select geo_lon as geo_lou from city)
...> select sqrt(power((geo_las - geo_lam),2) + power((geo_los - geo_lou),2)) As distance, city from (geo_las, geo_
los, geo_lam, geo_lou)
...> where city != 'Самара'
...> order by distance ASC limit 3;
0.00105299999999886|Заречный
0.0094843000000004|Каменка
0.0119931000000051|Елизово
sqlite>
```

Рисунок 6 – Задание 6

2.13. Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

```
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
sqlite>
```

Рисунок 7 – Задание 7 в формате csv с заголовками

```
sqlite> .separator |
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 8 – Задание 7 с разделителями «pipe»

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.

	Country (or dependency)	Population (2020)	Yearly Change	Net Change	Density (P/Km ²)	Land Area (Km ²)	Migrants (net)	Fert. Rate	Med. Age	Urban Pop %	World Share
1	China	1440297825	0.39 %	5540090	153	9388211	-348399.0	1.7	38	61 %	18.47 %
2	India	1382345085	0.99 %	13586631	464	2973190	-532687.0	2.2	28	35 %	17.70 %
3	United States	331341050	0.59 %	1937734	36	9147420	954806.0	1.8	38	83 %	4.25 %
4	Indonesia	274021604	1.07 %	2898047	151	1811570	-98955.0	2.3	30	56 %	3.51 %
5	Pakistan	231612385	2.00 %	4327022	287	770880	323270.0	2.6	32	35 %	2.62 %

Рисунок 9 – Выбранный датасет (population.csv)

1. Сформировал первый запрос.

```
sqlite> SELECT country, populati from population limit 5;
```

Country	Populati
China	1440297825
India	1382345085
United States	331341050
Indonesia	274021604
Pakistan	221612785

Рисунок 10 – Вывод информации о стране и населении

2. Сформировал второй запрос.

```
sqlite> SELECT country, populati FROM population WHERE populati BETWEEN 10000000 AND 12000000;
```

Country	Populati
Ethiopia	115434444
Philippines	109830324
Egypt	102659126
Burundi	11948454
Tunisia	11839918
Bolivia	11700207
Belgium	11598451
Haiti	11426356
Cuba	11325391
South Sudan	11216250
Dominican Republic	10866667
Czech Republic (Czechia)	10712481
Greece	10413936
Jordan	10220604
Portugal	10191409
Azerbaijan	10154978
Sweden	10110233
Eswatini	1162233
Kiribati	119760
Micronesia	115231
Grenada	112614
St. Vincent & Grenadines	111002
Aruba	106845
Tonga	105901
U.S. Virgin Islands	104398
Tuvalu	11817
Wallis & Futuna	11203
Nauru	10836

```
sqlite> _
```

Рисунок 11 – Выбрать страны, с населением от 100.000.000 до 120.000.000 человек

3. Сформировал третий запрос.

```
sqlite> SELECT country, density FROM population order by density desc limit 5;
```

Country	Density
Malaysia	99
Dominica	96
Syria	95
Cambodia	95
Kenya	94

```
sqlite> _
```

Рисунок 12 – Выбрать страну с плотностью населения (в процентах) и отсортировать по уменьшению

4. Сформировал четвертый запрос.

```
sqlite> select country, populati, yearly_change from population where yearly_change > "2 %" order by yearly_change desc limit 15;
```

Country	Populati	yearly_change
Niger	24346468	3.84 %
Bahrain	1711057	3.68 %
Equatorial Guinea	1410419	3.47 %
Uganda	45974931	3.32 %
Angola	33032075	3.27 %
DR Congo	90003954	3.19 %
Burundi	11948454	3.12 %
Falkland Islands	3497	3.05 %
Mali	20346106	3.02 %
Chad	16502877	3.00 %
Tanzania	60012400	2.98 %
Gambia	2427782	2.94 %
Mozambique	31398811	2.93 %
Zambia	18468257	2.93 %
Somalia	15965848	2.92 %

Рисунок 13 – Выбрать страны с изменением численности населения по годам выше 2%

5. Сформировал пятый запрос.

```
sqlite> select country, med_age from population WHERE fert_rate > "5" limit 10;
```

Country	Med_Age
Nigeria	18
DR Congo	17
Uganda	17
Angola	17
Niger	15
Burkina Faso	18
Mali	16
Chad	17
Somalia	17
Burundi	17

Рисунок 14 – Средний возраст здорового поколения в стране с учетом рождаемости и темпа роста в этих странах

6. Сформировал шестой запрос.

```
sqlite> select country from population where urban_pop > "95 %";
```

Country
Venezuela
Belgium
Hong Kong
Singapore
Kuwait
Uruguay
Qatar
Puerto Rico
Macao
Guadeloupe
U.S. Virgin Islands
Cayman Islands
Bermuda
Sint Maarten
Monaco
San Marino
Gibraltar
Palau
Anguilla
Nauru
Holy See

Рисунок 15 – Отобразить страны, где городское население занимает выше 95% самой страны

Контрольные вопросы:

1. Каково назначение реляционных баз данных и СУБД?

Главная функция СУБД – это управление данными (которые могут быть как во внешней, так и в оперативной памяти). СУБД обязательно поддерживает языки баз данных, а также отвечает за копирование и восстановление информации.

после каких-либо сбоев.

2. Каково назначение языка SQL?

Язык SQL предназначен для создания и изменения реляционных баз данных, а также извлечения из них данных. Другими словами, SQL – это инструмент, с помощью которого человек управляет базой данных.

3. Из чего состоит язык SQL?

Язык SQL состоит из операторов, инструкций и вычисляемых функций.

4. В чем отличие СУБД SQLite от клиент-серверных СУБД?

С помощью SQLite создаются базы данных, представляющие собой один кроссплатформенный текстовый файл. Файл базы данных, в отличие от SQLite, не встраивается в приложение, не становится его частью, он

существует отдельно. Так можно создать базу данных, пользуясь консольным `sqlite3`, после чего использовать ее в программе с помощью библиотеки `SQLite` языка программирования. При этом файл базы данных также хранится на локальной машине.

5. Как установить SQLite в Windows и Linux?

В Ubuntu установить `sqlite3` можно командой `sudo apt install sqlite3`. Для операционной системы Windows скачивают свой архив (`sqlite-tools-win32-*.zip`) и распаковывают.

6. Как создать базу данных SQLite?

С помощью `sqlite3` создать или открыть существующую базу данных можно двумя способами. Во-первых, при вызове утилиты `sqlite3` в качестве аргумента можно указать имя базы данных. Если БД существует, она будет открыта. Если ее нет, она будет создана и открыта.

7. Как выяснить в SQLite какая база данных является текущей? Выяснить, какая база данных является текущей, можно с помощью команды `.databases` утилиты `sqlite3`.

8. Как создать и удалить таблицу в SQLite?

Таблицы базы данных создаются с помощью директивы `CREATE TABLE` языка SQL. После `CREATE TABLE` идет имя таблицы, после которого в скобках перечисляются имена столбцов и их тип. Для удаления целой таблицы из базы данных используется директива `DROP TABLE`, после которой идет имя удаляемой таблицы.

9. Что является первичным ключом в таблице?

`PRIMARY KEY` – ограничитель, который заставляет СУБД проверять уникальность значения данного поля у каждой добавляемой записи.

10. Как сделать первичный ключ таблицы автоинкрементным?

Добавить `AUTOINCREMENT` в столбце при создании таблицы

11. Каково назначение инструкций `NOT NULL` и `DEFAULT` при создании таблиц?

Ограничитель `NOT NULL` используют, чтобы запретить оставление поля

пустым.

DEFAULT задает значение по умолчанию.

12. Каково назначение внешних ключей в таблице? Как создать внешний ключ в таблице?

С помощью внешнего ключа устанавливается связь между записями разных таблиц.

Чтобы включить поддержку внешних ключей в sqlite3, надо выполнить команду PRAGMA foreign_keys = ON. После этого

добавить в таблицу запись, в которой внешний ключ не совпадает ни с одним первичным из другой таблицы, не получится.

13. Как выполнить вставку строки в таблицу базы данных SQLite? С помощью оператора INSERT языка SQL выполняется вставка данных в таблицу.

14. Как выбрать данные из таблицы SQLite?

С помощью оператора SELECT осуществляется выборочный просмотр данных из таблицы.

15. Как ограничить выборку данных с помощью условия WHERE?

Условие WHERE используется не только с оператором SELECT, также с UPDATE и DELETE. С помощью WHERE определяются строки, которые будут выбраны, обновлены или удалены. По сути это фильтр.

16. Как упорядочить выбранные данные?

При выводе данных их можно не только фильтровать с помощью WHERE, но и сортировать по возрастанию или убыванию с помощью оператора ORDER BY.

17. Как выполнить обновление записей в таблице SQLite?

UPDATE ... SET – обновление полей записи

18. Как удалить записи из таблицы SQLite?

DELETE FROM – удаление записей таблицы

19. Как сгруппировать данные из выборки из таблицы SQLite?

В SQL кроме функций агрегирования есть оператор GROUP BY,

который выполняет группировку записей по вариациям заданного поля.

20. Как получить значение агрегатной функции (например: минимум, максимум, количество записей и т. д.) в выборке из таблицы SQLite? Для этих целей в языке SQL предусмотрены различные функции агрегирования данных. Наиболее используемые – count(), sum(), avr(), min(), max().

21. Как выполнить объединение нескольких таблиц в операторе SELECT?

После FROM указываются обе сводимые таблицы через JOIN. В данном случае неважно, какую указывать до JOIN, какую после. После ключевого слова ON записывается условие сведения. Условие сообщает, как соединять строки разных таблиц.

22. Каково назначение подзапросов и шаблонов при работе с таблицами SQLite?

Шаблоны реализуют поиск по таблице, если неизвестно полное название данных в строке.

Подзапросы помогают уменьшить работу путём создания дополнительного запроса внутри основного.

23. Каково назначение представлений VIEW в SQLite?

Бывает удобно сохранить результат выборки для дальнейшего использования. Для этих целей в языке SQL используется оператор CREATE VIEW, который создает представление – виртуальную таблицу. В эту виртуальную таблицу как бы сохраняется результат запроса.

24. Какие существуют средства для импорта данных в SQLite?

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

25. Каково назначение команды .schema ?

Показывает какие столбцы есть в таблице, тип их данных и прочие свойства.

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

```
select federal_district as district, count(*) as city_count from city group by 1
```

order by 2 desc;

27. Каково назначение "табличных выражений" в SQLite?

Выражение with history as (...) создает именованный запрос. Название — history , а содержание — селект в скобках (век основания для каждого города). К history можно обращаться по имени в остальном запросе, что мы и делаем.

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

.mode csv

29. Какие еще форматы для экспорта данных Вам известны?

.mode list

.mode json

Вывод: в ходе выполнения лабораторной работы были исследованы базовые возможности системы управления базами данных SQLite3.