

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №1

Основы работы с SQLite3

По дисциплине «Технологии программирования и алгоритмизация»

Выполнил студент группы ИВТ-б-о-21-1

Харченко Б.Р. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

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

Ход работы:

1. Создал репозиторий и клонировал его на компьютер.
2. Выполнил общие задания.
3. Повторил команду.

```
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. Задание 1

4. 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

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

```
sqlite> select max(length(city)) from city;
25
Run Time: real 0.015 user 0.015625 sys 0.000000
sqlite> _
```

Рисунок 3. Задание 3

6. 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
sqlite> _
```

Рисунок 4. Задание 4

7. 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 = 'Сибирский' GROUP BY timezone ORDER BY timezone ASC;
```

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

Рисунок 5. Задание 5

8. 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.

distance	city
0.00105299999999886	Заречный
0.0094843000000004	Каменка
0.01199310000000051	Елизово

Рисунок 6. Задание 6

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

```
sqlite> SELECT timezone, count(Distinct city) AS city_count FROM city GROUP BY timezone ORDER BY timezone ASC;
```

timezone	city_count
UTC+10	22
UTC+11	17
UTC+12	6
UTC+2	22
UTC+3	652
UTC+4	66
UTC+5	173
UTC+6	6
UTC+7	86
UTC+8	28
UTC+9	31

Рисунок 7. Задание 7 в формате box

```
sqlite> SELECT
UTC+10,22
UTC+11,17
UTC+12,6
UTC+2,22
UTC+3,652
UTC+4,66
UTC+5,173
UTC+6,6
UTC+7,86
UTC+8,28
UTC+9,31
sqlite> █
```

Рисунок 8. Задание 7 в формате csv

```
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 9. 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.

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
1	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu
2	15	8	350	165	3693	11.5	70	1	buick skylark 320
3	18	8	318	150	3436	11	70	1	plymouth satellite
4	16	8	304	150	3433	12	70	1	amc rebel sst
5	17	8	302	140	3449	10.5	70	1	ford torino
6	15	8	429	198	4341	10	70	1	ford galaxie 500
7	14	8	454	220	4354	9	70	1	chevrolet impala

Рисунок 10. Выбранный датасет

10. Сформировал первый запрос. Найти машины с расходом не более 10 миль на галлон.

```
sqlite> select mpg, car_name from mpg where mpg < 15  
...> order by mpg ASC;  
mpg,car_name  
10,"ford f250"  
10,"chevy c20"  
11,"dodge d200"  
11,"mercury marquis"  
11,"chevrolet impala"  
11,"oldsmobile omega"  
12,"dodge monaco (sw)"  
12,"oldsmobile delta 88 royale"  
12,"mercury marquis brougham"  
12,"buick electra 225 custom"  
12,"ford country"  
12,"oldsmobile vista cruiser"  
13,"ford country squire (sw)"  
13,"pontiac safari (sw)"  
13,"chevrolet impala"  
13,"buick lesabre custom"  
13,"chrysler newport royal"  
13,"chevrolet chevelle concours (sw)"  
13,"ford gran torino (sw)"  
13,"buick century 350"  
13,"chevrolet malibu"  
13,"chevrolet caprice classic"  
13,"ford ltd"  
13,"chrysler new yorker brougham"  
13,"amc ambassador brougham"
```

Рисунок 11. Первый запрос

11. Сформировал второй запрос. Сколько машин имеют мощность больше 160 Кв и 8 цилиндров.

```
sqlite> select count(car_name) from mpg where horsepower > 160 and cylinders = 8  


|                 |
|-----------------|
| count(car_name) |
| 37              |


```

Рисунок 12. Второй запрос

12. Сформировал третий запрос. Найти средний вес автомобилей 70-го года выпуска.

```
sqlite> select avg(weight) as AverageWeight from mpg where model_year = 70;
AverageWeight
3372.79310344828
```

Рисунок 13. Третий запрос

13. Сформировал четвёртый запрос. Найти машины, выпущенные между 74 и 76 годом.

```
sqlite> select car_name, model_year from mpg where model_year between 74 and 76;
car_name,model_year
"plymouth duster",74
"ford maverick",74
"amc hornet",74
"chevrolet nova",74
"datsun b210",74
"ford pinto",74
"toyota corolla 1200",74
"chevrolet vega",74
"chevrolet chevelle malibu classic",74
"amc matador",74
"plymouth satellite sebring",74
"ford gran torino",74
"buick century luxus (sw)",74
"dodge coronet custom (sw)",74
```

Рисунок 14. Четвертый запрос

14. Сформировал пятый запрос. Вывести список машин с расходом меньше 20 миль на галлон и ускорением меньше 10 м/с².

```
sqlite> select car_name, acceleration, mpg from mpg where mpg < 20 and acceleration < 12;
car_name,acceleration,mpg
"buick skylark 320",11.5,15
"plymouth satellite",11,18
"ford torino",10.5,17
"ford galaxie 500",10,15
"pontiac catalina",10,14
"dodge challenger se",10,15
"buick estate wagon (sw)",10,14
"pontiac catalina brougham",11.5,14
"dodge monaco (sw)",11.5,12
"amc ambassador sst",11.5,17
"mercury marquis",11,11
"amc matador",11.5,14
"mercury marquis brougham",11.5,12
"chrysler new yorker brougham",11,13
"buick electra 225 custom",11,12
"amc ambassador brougham",11,13
"dodge dart custom",11,15
"oldsmobile omega",11,11
"pontiac catalina",11.5,16
"pontiac grand prix lj",11.1,16
"chevrolet monte carlo landau",11.4,15.5
```

Рисунок 15. Пятый запрос

15. Сформировал шестой запрос. Найти 5 автомобилей, которые имеют 6 цилиндров и разгон от 10 до 15.

```
sqlite> select car_name, cylinders, acceleration from mpg where cylinders = 6 and acceleration between 10 and 15
...> order by acceleration
...> limit 5;
"chevrolet citation",6,11.3
"datsun 280-zx",6,11.4
"chevrolet citation",6,12.6
"toyota cressida",6,12.6
"oldsmobile omega brougham",6,12.9
```

Рисунок 16. Шестой запрос

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 исследовал базовые возможности системы управления базами данных SQLite3.