

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

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.20

Основы работы с SQLite3

По дисциплине «Теории программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р. А. _____

(подпись)

Ставрополь 2022

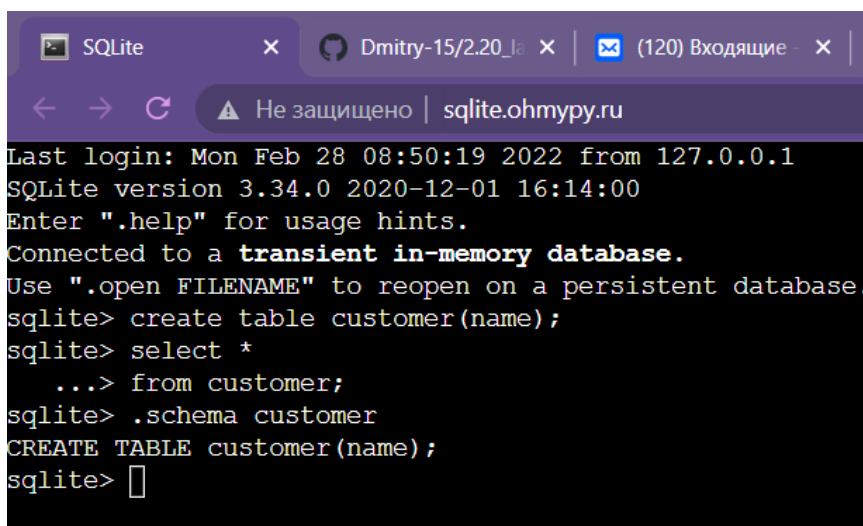
Цель работы: исследовать базовые возможности системы управления базами данных SQLite3.

Ход работы

1. Создал новый собственный репозиторий. Ссылка на репозиторий: https://github.com/Dmitry-15/2.20_laba.
2. С помощью команды `git clone` клонировал удаленный репозиторий на свой ПК. Дополнил файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
3. После ознакомления с теоретическим материалом приступил к выполнению заданий.

Задание 1

1. Условие: выполните в песочнице команды:
`create table customer(name);`
`select *`
`from customer;`
`.schema customer`



```
SQLite x Dmitry-15/2.20_laba x (120) Входящие x
← → ↻ Не защищено | sqlite.ohmypy.ru
Last login: Mon Feb 28 08:50:19 2022 from 127.0.0.1
SQLite version 3.34.0 2020-12-01 16:14:00
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 – Результат выполнения первого задания

Задание 2

1. Условие: с помощью команды `.help` найти (команда `.timer`) в песочнице команду, которая отвечает за вывод времени выполнения запроса. Ввести команду:
`sqlite> .timer on`

```
sqlite> select count(*) from city;
```

```
1117
```

```
Run Time: real 0.000 user 0.000106 sys 0.000069
```

```
sqlite> .timer on
sqlite> select count(*) from city;
1117
Run Time: real 0.000 user 0.000000 sys 0.000088
```

Рисунок 2 – Результат выполнения второго задания

Задание 3

1. Условие: загрузите файл city.csv в песочнице: `.import --csv city.csv city`. Затем выполните такой запрос: `select max(length(city)) from city;` и посмотреть какое число он вернет.

```
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
sqlite> █
```

Рисунок 3 – Результат выполнения третьего задания

Задание 4

1. Условие: загрузите файл city.csv в песочнице с помощью команды `.import`, но без использования опции `--csv`. Эта опция появилась только в недавней версии SQLite (3.32, май 2020), так что полезно знать способ, подходящий для старых версий. Вам поможет команда `.help import`. Всего должно получиться две команды: `do_something` и `.import city.csv city`, вместо `do_something` ввести нужную команду (`.mode csv`).

```
sqlite> .help import
.import FILE TABLE      Import data from FILE into TABLE
Options:
  --ascii                Use \037 and \036 as column and row separators
  --csv                  Use , and \n as column and row separators
  --skip N               Skip the first N rows of input
  -v                     "Verbose" - increase auxiliary output
Notes:
  * If TABLE does not exist, it is created. The first row of input
    determines the column names.
  * If neither --csv or --ascii are used, the input mode is derived
    from the ".mode" output mode
  * If FILE begins with "|" then it is a command that generates the
    input text.
sqlite> .mode csv
sqlite> .import city.csv city
```

Рисунок 4 – Результат выполнения четвертого задания

Задание 5

1. Условие: 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. Укажите в ответе значение `city_count` для `timezone = UTC+5`.

2. Для этого ввел команду: `select timezone, count(city) as city_count from city where federal_district = 'Приволжский' or federal_district = 'Сибирский' group by timezone order by timezone asc;`

```
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 – Результат выполнения пятого задания

Задание 6

1. Условие: напишите в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару.

2. Для этого ввел команду: `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;`

```
sqlite> with geo_las as (select geo_lat as geo_las from city where city = 'Самапа'),
...> geo_los as (select geo_lon as geo_los from city where city = 'Самапа'),
...> 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)))
...> ^Cas distance, city from (geo_las ,geo_los ,geo_lam, geo_lou)
...> where city != 'Самапа' order by distance asc limit 3;
```

distance	city
0.001052999999999886	Заречный
0.0094843000000004	Каменка
0.01199310000000051	Елизово

Рисунок 6 – Результат выполнения шестого задания

Задание 7

1. Условие: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию. Затем выполните этот же запрос, но так, чтобы результат был в формате CSV, с заголовками, с разделителем «pipe» |.

Для этого ввел команду: `select timezone, count(distinct city) as city_count from city group by timezone order by timezone asc;`

```
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 – Результат выполнения седьмого задания

```

sqlite> .mode csv
sqlite> .headers on
sqlite> .separator |
sqlite> select timezone, count(distinct city) as city_count from city group by timez
one 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 8 – Результат в формате CSV, с заголовками и разделителем «pipe» |





Индивидуальное задание

1. Условие: 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.

2. Выбрал датасет связанный с темой Covid-19.

15 lines (15 sloc) | 923 Bytes

RawBlame



Q Search this file...

	Country/Other	Cases in the last 7 days	Cases in the preceding 7 days	Weekly Case % Change	Cases in the last 7 days/1M pop	Deaths in the last 7 days	Deaths in the preceding 7 days	Weekly Death % Cha
1	Argentina	18322	36321	-50	399	415	307	35
3	Bolivia	1595	2312	-31	133	5	9	-44
4	Brazil	211934	261708	-19	985	1665	2122	-22
5	Chile	56504	91843	-38	2913	539	803	-33
6	Colombia	2512	4312	-42	48	121	155	-22
7	Ecuador	3284	4629	-29	181	8	63	-87
8	Falkland Islands	1	0	100	274	0	0	0
9	French Guiana	323	251	29	1035	0	0	0
10	Guyana	40	50	-20	50	0	2	-100
11	Paraguay	807	1216	-34	111	28	48	-42
12	Peru	4006	4901	-18	119	263	319	-18
13	Suriname	222	174	28	373	1	2	-50
14	Uruguay	8768	10937	-20	2509	13	43	-70
15	Venezuela	443	1260	-65	16	11	10	10

Рисунок 9 – Выбранный датасет

3. Затем сформировал 6 запросов к таблице.

```
sqlite> .mode box
sqlite> .headers on
sqlite> select Country, CasesLast7d from covid where CasesLast7d between 2500 and 3000;
```

Country	CasesLast7d
Colombia	2512

Рисунок 10 – Первый запрос

```
sqlite> select Country, DeathsLast7d from covid where DeathsLast7d == 0 limit 2;
```

Country	DeathsLast7d
Falkland Islands	0
French Guiana	0

Рисунок 11 – Второй запрос

```
sqlite> select Country, Population from covid order by Population asc limit 10;
```

Country	Population
Bolivia	11948668
Ecuador	18106806
Chile	19400316
Brazil	215175647
Venezuela	28296430
French Guiana	312076
Peru	33766958
Uruguay	3494576
Falkland Islands	3655
Argentina	45913837

Рисунок 12 – Третий запрос

```
sqlite> select Country, CasesPreceding7d, Population from covid where CasesPreceding7d between 4000 and 4999 order by Population desc limit 3;
```

Country	CasesPreceding7d	Population
Colombia	4312	51821865
Peru	4901	33766958
Ecuador	4629	18106806

Рисунок 13 – Четвертый запрос

```
sqlite> select Country, WeeklyCasePerChange, WeeklyDeathPerChange from covid where WeeklyCasePerChange <= 0 and WeeklyDeathPerChange != 0 limit 8;
```

Country	WeeklyCasePerChange	WeeklyDeathPerChange
Argentina	-50	35
Bolivia	-31	-44
Brazil	-19	-22
Chile	-38	-33
Colombia	-42	-22
Ecuador	-29	-87
Guyana	-20	-100
Paraguay	-34	-42

Рисунок 14 – Пятый запрос

```
sqlite> select Country, CasesLast7d1MPop, CasesPreceding7d, Population from covid where CasesLast7d1MPop between 0 and 250 and CasesPreceding7d != 0 order by Population desc;
```

Country	CasesLast7d1MPop	CasesPreceding7d	Population
Paraguay	111	1216	7284217
Peru	119	4901	33766958
French Guiana	1035	251	312076
Venezuela	16	1260	28296430
Ecuador	181	4629	18106806
Bolivia	133	2312	11948668

Рисунок 15 – Шестой запрос

4. Далее выгрузил результаты запросов в файлы формата CSV и JSON.













 request1.csv	03.04.2022 23:15	Файл Microsoft Ex...	1 КБ
 request1.json	03.04.2022 23:16	Файл "JSON"	1 КБ
 request2.csv	03.04.2022 23:43	Файл Microsoft Ex...	1 КБ
 request2.json	03.04.2022 23:44	Файл "JSON"	1 КБ
 request3.csv	03.04.2022 23:46	Файл Microsoft Ex...	1 КБ
 request3.json	03.04.2022 23:47	Файл "JSON"	1 КБ
 request4.csv	03.04.2022 23:57	Файл Microsoft Ex...	1 КБ
 request4.json	03.04.2022 23:57	Файл "JSON"	1 КБ
 request5.csv	04.04.2022 10:10	Файл Microsoft Ex...	1 КБ
 request5.json	04.04.2022 10:10	Файл "JSON"	1 КБ
 request6.csv	04.04.2022 10:18	Файл Microsoft Ex...	1 КБ
 request6.json	04.04.2022 10:18	Файл "JSON"	1 КБ

Рисунок 16 – Выгрузка результатов запросов в CSV и JSON

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 citygroup 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 успешно были исследованы базовые возможности системы управления базами данных SQLite3.