

**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 «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-20-1

Хашиев Х.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2021

Цель работы: приобретение навыков с работой SQLite3 с помощью языка программирования Python версии 3.x.

Ход работы: Примеры

<https://github.com/Mirror-Shard/L2.20>

Задание 1

1. Изучил методические указания и приступил к выполнению заданий:

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

Рисунок 1 – Первое задание

2. Запустил программу и ввёл информацию о двух студентах двумя командами:

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


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

  
Run Time: real 0.001 user 0.000167 sys 0.000071
```

Рисунок 2 – Второе задание

3. Результат работы:

```
sqlite> .mode csv  
sqlite> .import city.csv city  
sqlite> select count(*) from city;  
2235
```

Рисунок 3 – Третье задание

4. Написал программу в соответствии с заданием и добавил несколько функций с помощью парсера:

```
sqlite> select max(length(city)) from city;  
25
```

Рисунок 4 – Четвёртое задание

5. Затем вывел значения на экран, программа работает успешно:

```
sqlite> select timezone, count(*) as city_count from city where federal_district in ('Приволжский', 'Сибирский') and timezone == "UTC+5" group by timezone;
```

timezone	city_count
UTC+5	116

Рисунок 5 – Пятое задание

```
sqlite> select timezone, count(*) as city_count from city group by timezone order by 2 asc;
```

timezone	city_count
1	1
UTC+12	12
UTC+6	12
UTC+11	34
UTC+10	44
UTC+2	44
UTC+8	56
UTC+9	62
UTC+4	132
UTC+7	172
UTC+5	346
UTC+3	1320

Рисунок 6 - Шестое задание

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.

1. Для этого скачал готовый датасет и произвёл 5 запросов к полученной базе данных:

```
sqlite> .once zapros1.csv
sqlite> select "Student ID", "Degree of difficulty of assignments", "Well versed with the subject" from stud where "Degree of difficulty of assignments" > 5 and "Well versed with the subject" > 7
...> order by "Degree of difficulty of assignments", "Well versed with the subject";
```

Рисунок 7 – Первый запрос

```
sqlite> .once zapros2.csv
sqlite> select "Provides support for students going above and beyond", count(*) as "Number of students" from stud
...> group by "Provides support for students going above and beyond"
...> order by 2;
```

Рисунок 8 – Второй запрос

```
sqlite> .once zapros3.csv
sqlite> select "Student ID", "Structuring of the course" from stud
...> where "Structuring of the course" == 10 order by 1;
```

Рисунок 9 – Третий запрос

```
sqlite> .once zapros4.csv
sqlite> select count(*) as "Numbers of student", "Course recommendation based on relevance" from stud
...> where "Course recommendation based on relevance" > 6
...> group by 2
...> order by 2 desc;
```

Рисунок 10 – Четвёртый запрос

```
sqlite> .once zapros5.csv
sqlite> select "Student ID" from stud
...> where "Well versed with the subject" > 400 and "Explains concepts in an understandable way" > 4 and "Use of presentations" > 4
...> and "Degree of difficulty of assignments" > 4 and "Solves doubts willingly" > 4 and "Structuring of the course" > 4
...> and "Provides support for students going above and beyond" > 4 and "Course recommendation based on relevance" > 4
...> limit 10;
```

Рисунок 11 – Пятый 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 работы приобрёл навыки в SQLite3 при написании программ с помощью языка программирования Python версии 3.