

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**  
**дисциплины «Анализ данных»**

Выполнил:  
Данилецкий Дмитрий Витальевич  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А., канд. технических  
наук, доцент кафедры  
инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

## Тема. Основы работы с SQLite3

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

### Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.

2. Дополнил файл .gitignore необходимыми правилами.

3. Организовал созданный репозиторий в соответствии с необходимыми требованиями.

4. Добавил в файл README.md информацию о группе и ФИО студента, выполняющего лабораторную работу.

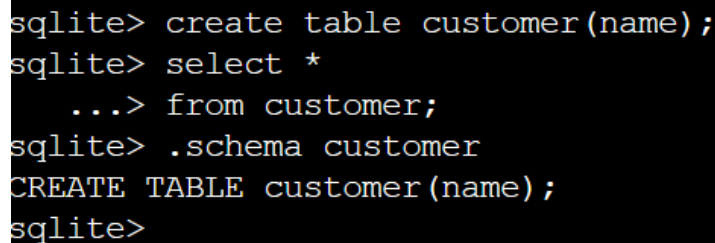
5. Решил задачу: выполнил в песочнице команды:

```
create table customer(name);
```

```
select *
```

```
from customer;
```

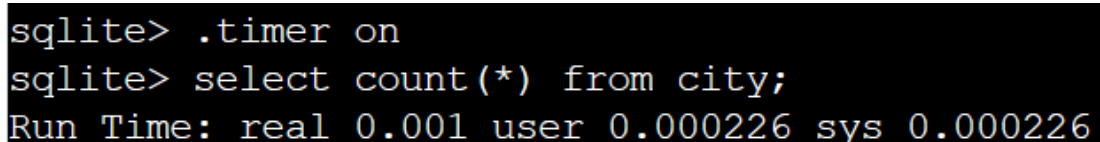
```
.schema customer
```



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

Рисунок 1. Результат работы программы из задачи 1

6. Решил задачу: с помощью команды .help нашёл в песочнице команду, которая отвечает за вывод времени выполнения запроса.



```
sqlite> .timer on  
sqlite> select count(*) from city;  
Run Time: real 0.001 user 0.000226 sys 0.000226
```

Рисунок 2. Результат работы программы из задачи 2

7. Решил задачу: загрузил файл city.csv

```

sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
Run Time: real 0.001 user 0.000568 sys 0.000213
sqlite> 

```

Рисунок 3. Результат работы программы из задачи 3

8. Решил задачу: загрузил файл city.csv в песочнице с помощью команды .import , но без использования опции --csv .

```

sqlite> .separator
Usage: .separator COL ?ROW?
sqlite> .separator ,
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
Run Time: real 0.002 user 0.001739 sys 0.000000
sqlite> 

```

Рисунок 4. Результат работы программы из задачи 4

9. Решил задачу: написал в песочнице запрос, который посчитает количество городов для каждого часового пояса в Сибирском и Приволжском федеральных округах. Вывел столбцы timezone и city\_count, отсортировал по значению часового пояса:

```

sqlite> .import --csv city.csv city
sqlite> SELECT
...> DISTINCT timezone AS timee,
...> COUNT() AS count_city
...> FROM city
...> GROUP BY timee
...> ORDER BY timee DESC;

```

timee	count_city
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

Рисунок 4. Результат работы программы из задачи 5

10. Решил задачу: написал в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару.

```
sqlite> .import --csv city.csv city
sqlite> WITH samara AS (
...>   SELECT geo_lat AS lat, geo_lon AS lon
...>   FROM city
...>   WHERE city = "Самара"
...> )
...> SELECT
...>   city,
...>   SQRT(
...>     POW(geo_lon - lon, 2) + POWER(geo_lat - lat, 2)
...>   ) * 69.09 AS euclidean_distance
...> FROM
...>   city, samara
...> WHERE
...>   euclidean_distance > 0
...> ORDER BY
...>   euclidean_distance
...> LIMIT 3;
Новокуйбышевск|12.8298063265514
Чапаевск|24.7389598092284
Кинель|36.4840951529617
sqlite> 
```

Рисунок 5. Результат работы программы из задачи 6

10. Решил задачу: написал в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

```
sqlite> .mode csv
sqlite> .headers ON
sqlite> .separator |
sqlite> SELECT
...>   timezone AS timezone,
...>   COUNT(*) AS city_count
...> FROM
...>   city
...> GROUP BY
...>   timezone
...> ORDER BY
...>   city_count 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 6. Результат работы программы из задачи 7

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.

```
sqlite> .once C:/Users/slime/Downloads/2/1.json
sqlite> SELECT Country, MAX(Stars) AS Max_Stars, Variety
...> FROM ramen_reviews
...> GROUP BY Country;
Parse error: no such table: ramen_reviews
sqlite> .once C:/Users/slime/Downloads/2/1.json
sqlite> SELECT Country, MAX(Stars) AS Max_Stars, Variety
...> FROM ramen
...> GROUP BY Country;
sqlite> .once C:/Users/slime/Downloads/2/2.json
sqlite> SELECT * FROM ramen LIMIT 5;
sqlite> .once C:/Users/slime/Downloads/2/3.json
sqlite> SELECT Brand, AVG(Stars) AS Avg_Stars FROM ramen GROUP BY Brand;
sqlite> .once C:/Users/slime/Downloads/2/4.json
sqlite> SELECT Brand, COUNT(*) AS Review_Count
...> FROM ramen
...> WHERE Stars > 4
...> GROUP BY Brand;
sqlite> .once C:/Users/slime/Downloads/2/5.json
sqlite> SELECT Style, MAX(Stars) AS Max_Stars
...> FROM ramen
...> GROUP BY Style;
sqlite> .once C:/Users/slime/Downloads/2/6.json
sqlite> SELECT Brand, COUNT(*) AS Review_Count
...> FROM ramen_reviews
...> GROUP BY Brand
...> ORDER BY Review_Count DESC
...> LIMIT 10;
Parse error: no such table: ramen_reviews
sqlite> .once C:/Users/slime/Downloads/2/6.json
sqlite> SELECT Brand, COUNT(*) AS Review_Count
...> FROM ramen
...> GROUP BY Brand
...> ORDER BY Review_Count DESC
...> LIMIT 10;
```

Рисунок 7. Запросы к БД

### Контрольные вопросы

#### 1. Каково назначение реляционных баз данных и СУБД?

Слово "реляция" происходит от "relation", то есть "отношение". Это означает, что в РБД существуют механизмы установления связей между таблицами. Делается это с помощью так называемых первичных и внешних ключей.

#### 2. Каково назначение языка SQL?

SQL – это язык программирования декларативного типа. В отличие от привычных нам процедурных языков, в которых есть условия, циклы и функции, в декларативных языках подобных алгоритмических конструкций почти нет. Декларативные выражения представляют собой скорее запросы, описание того, что хочет получить человек.

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. При этом ключевыми операциями являются создание таблиц, добавление записей в таблицы, изменение и удаление записей, выборка записей из таблиц, изменение структуры таблиц.

### 3. Из чего состоит язык SQL?

Сам язык SQL состоит из операторов, инструкций и вычисляемых функций. Зарезервированные слова, которыми обычно выступают операторы, принято писать заглавными буквами. Однако написание их не прописными, а строчными буквами к ошибке не приводит.

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

SQLite – это система управления базами данных, отличительной особенностью которой является ее встраиваемость в приложения. Это значит, что большинство СУБД являются самостоятельными приложениями, взаимодействие с которыми организовано по принципу клиент-сервер. Программа-клиент посылает запрос на языке SQL, СУБД, которая в том числе может находиться на удаленном компьютере, возвращает результат запроса.

В свою очередь SQLite является написанной на языке C библиотекой, которую динамически или статически подключают к программе. Для большинства языков программирования есть свои привязки (API) для библиотеки SQLite. Так в Python СУБД SQLite импортируют командой `import sqlite3`. Причем модуль `sqlite3` входит в стандартную библиотеку языка и не требует отдельной установки.

### 5. Как установить SQLite в Windows и Linux?

Для операционной системы Windows скачивают свой архив (`sqlite-tools win32-*.zip`) и распаковывают. Далее настраивают путь к каталогу, добавляя адрес каталога к переменной `PATH` (подобное можно сделать и в Linux).

## 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.

```
$ sqlite3 your.db
```

Во вторых, работая в самой программе, можно выполнить команду `.open your.db`

## 7. Как выяснить в SQLite какая база данных является текущей?

Выяснить, какая база данных является текущей, можно с помощью команды `.databases` утилиты `sqlite3`. Если вы работаете с одной БД, а потом открываете другую, то текущей становится вторая БД.

## 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?

Первичный ключ в таблице базы данных — это уникальный идентификатор для каждой строки таблицы. Он гарантирует уникальность записи и используется для создания связей между таблицами (внешних ключей).

В архитектуре реляционной базы данных, первичный ключ может состоять из одного столбца (простой или одиночный ключ) или может комбинировать несколько столбцов (составной ключ), если уникальность может быть гарантирована только их комбинацией. Значения первичного ключа должны быть уникальными и не могут быть NULL.

## 10. Как сделать первичный ключ таблицы автоинкрементным?

Если нам не важно, какие конкретно идентификаторы будут записываться в поле `_id`, а важна только уникальность поля, следует назначить полю еще один ограничитель – автоинкремент – `AUTOINCREMENT`.

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

Ограничитель `NOT NULL` используют, чтобы запретить оставление поля пустым. По умолчанию, если поле не является первичным ключом, в него можно не помещать данные. В этом случае полю будет присвоено значение `NULL`. В случае `NOT NULL` вы не сможете добавить запись, не указав значения соответствующего поля.

Однако, добавив ограничитель `DEFAULT`, вы сможете не указывать значение. `DEFAULT` задает значение по умолчанию. В результате, когда данные в поле не передаются при добавлении записи, поле заполняется тем, что было указано 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. Внешний ключ в одной таблице для другой является первичным. Внешние ключи не обязаны быть уникальными. В одной таблице, может быть, несколько внешних ключей, при этом каждый будет устанавливать связь со своей таблицей, где он является первичным.

13. Как выполнить вставку строки в таблицу базы данных SQLite?

С помощью оператора `INSERT` языка SQL выполняется вставка данных в таблицу.

```
INSERT INTO <table_name>
(<column_name1>, <column_name2>, ...)
VALUES
(<value1>, <value2>, ...);
```

14. Как выбрать данные из таблицы SQLite?



С помощью оператора SELECT.

15. Как ограничить выборку данных с помощью условия WHERE?

WHERE позволяет задавать условие, согласно которому отображаются только удовлетворяющие ему строки.

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.

ASC – сортировка от меньшего значения к большему.

DESC – сортировка от большего значения к меньшему.

17. Как выполнить обновление записей в таблице SQLite?

UPDATE ... SET – обновление полей записи

UPDATE имя\_таблицы

SET имя\_столбца = новое\_значение

WHERE условие;

18. Как удалить записи из таблицы SQLite?

DELETE FROM – удаление записей таблицы

DELETE FROM имя\_таблицы WHERE условие;

19. Как сгруппировать данные из выборке 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 CREATEVIEW, 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?

Группировка – GROUP BY, сортировка – ORDER BY.

27. Каково назначение "табличных выражений" в SQLite?

Табличные выражения в SQLite используются для структурирования запросов и улучшения их читаемости и гибкости. Табличные выражения можно считать временными наборами записей, которые доступны в пределах одного запроса.

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

`.mode csv` или `.mode json`

`.once <имя файла>` или `.output`

любой запрос

В результате вывод запроса уйдет в выбранный файл.

29. Какие еще форматы для экспорта данных Вам известны?

.mode list, .mode markdown, .mode html ...

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