

Северо-Кавказский федеральный университет
Институт математики и информационных технологий

ОТЧЕТ
о выполнении лабораторной работы №7
по дисциплине
«Основы Программной Инженерии»

Выполнил:

Ботвинкин Никита Сергеевич

студент 2 курса, ПИЖ-б-о-21-1 группы
бакалавриата «Программная инженерия»
очной формы обучения

Ставрополь, 2023

СКРИНШОТЫ ВЫПОЛНЕНИЯ ЗАДАЧ

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

Рисунок 7.1 – Задача №1

```
sqlite> .timer on  
sqlite> select count(*) from city;  
Run Time: real 0.000 user 0.000133 sys 0.000133
```

Рисунок 7.2 – Задача №2

```
sqlite> .import --csv city.csv city  
sqlite> select max(length(city)) from city;  
25  
Run Time: real 0.001 user 0.000913 sys 0.000000
```

Рисунок 7.3 – Задача №3

```
sqlite> .mode csv  
sqlite> .import city.csv city
```

Рисунок 7.4 – Задача №4

```
sqlite> select
...> timezone as Time_zone,
...> count(*) as counter
...> from city
...> where
...> timezone in ('UTC+3','UTC+4','UTC+5','UTC+6','UTC+7','UTC+8')
...> group by 1
...> order by 1;
```

Time_zone	counter
UTC+3	660
UTC+4	66
UTC+5	173
UTC+6	6
UTC+7	86
UTC+8	28

Рисунок 7.5 – Задача 5

```
sqlite> select
...> city,
...> (sqrt((((53.2001 - geo_lat)*(53.2001 - geo_lat))) + (( 50.15 - geo_lon)*(50.15 - geo_lon)))) as
distance
...> from city
...> where city <> 'Самара'
...> group by 1
...> order by 2 asc
...> limit 3;
```

city	distance
Новокуйбышевск	0.225918015730237
Чапаевск	0.391689211983949
Кинель	0.484829574087433

Рисунок 7.6 – Задача 6

```
sqlite> select
...> timezone,
...> count(*) as 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 7.7 – Задача 7.1

```
sqlite> .mode csv
sqlite> .headers on
sqlite> .separator |
sqlite> select
...> timezone,
...> count(*) as 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 7.8 – Задача 7.2

ПРИМЕРЫ

```
sqlite> .mode box
sqlite> create table city (id integer primary key, name text);
sqlite> insert into city (name) values ('Москва'), ('Санкт-Петербург'), ('Тамбов');
sqlite> select * from city;
```

id	name
	Москва
	Санкт-Петербург
	Тамбов

Рисунок 7.9 – Создание таблицы

```
sqlite> CREATE TABLE pages (
...> _id INTEGER PRIMARY KEY AUTOINCREMENT,
...> title TEXT,
...> url TEXT,
...> theme INTEGER,
...> num INTEGER);
sqlite> .schema pages
CREATE TABLE pages (
_id INTEGER PRIMARY KEY AUTOINCREMENT,
title TEXT,
url TEXT,
theme INTEGER,
num INTEGER);
sqlite> PRAGMA TABLE_INFO (pages);
```

cid	name	type	notnull	dflt_value	pk
0	_id	INTEGER	0		1
1	title	TEXT	0		0
2	url	TEXT	0		0
3	theme	INTEGER	0		0
4	num	INTEGER	0		0

Рисунок 7.10 – Автоинкремент

```

sqlite> CREATE TABLE pages (
...> _id INTEGER PRIMARY KEY AUTOINCREMENT,
...> title TEXT,
...> url TEXT NOT NULL,
...> theme INTEGER NOT NULL,
...> num INTEGER NOT NULL DEFAULT 0);
sqlite> .schema pages
CREATE TABLE pages (
_id INTEGER PRIMARY KEY AUTOINCREMENT,
title TEXT,
url TEXT NOT NULL,
theme INTEGER NOT NULL,
num INTEGER NOT NULL DEFAULT 0);
sqlite> PRAGMA TABLE_INFO(pages);
0|_id|INTEGER|0||1
1|title|TEXT|0||0
2|url|TEXT|1||0
3|theme|INTEGER|1||0
4|num|INTEGER|1|0|0

```

Рисунок 7.11 – NOT NULL и DEFAULT

```

sqlite> .help .mode
.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
  --schema S             Target table to be S.TABLE
  -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.
.mode MODE ?OPTIONS?     Set output mode
MODE is one of:
  ascii                  Columns/rows delimited by 0x1F and 0x1E
  box                    Tables using unicode box-drawing characters
  csv                    Comma-separated values
  column                 Output in columns. (See .width)
  html                   HTML <table> code
  insert                 SQL insert statements for TABLE
  json                   Results in a JSON array
  line                   One value per line
  list                   Values delimited by "|"
  markdown               Markdown table format
  qbox                   Shorthand for "box --width 60 --quote"
  quote                  Escape answers as for SQL

```

Рисунок 7.12 – Команда .help

```

sqlite> .mode column
sqlite> .header on
sqlite> SELECT * FROM pages;

```

_id	title	url	theme	num
1	What is Information	information	1	1

Рисунок 7.13 – Заголовки

```

sqlite> .import --csv city.csv city
sqlite> .mode box
sqlite> select count(*) from city;

```

count (*)
1117

Рисунок 7.14 – Команда .import

```

sqlite> select address
...> from city
...> where city like '%Красный%';

```

address
Ростовская обл, г Красный Сулин
Саратовская обл, г Красный Кут
Тверская обл, г Красный Холм

Рисунок 7.15 – Фильтрация

```

sqlite> with history as (
...> select
...> (foundation_year/100)+1 as century
...> from city
...> )
...> select
...> century || '-й век' as dates,
...> count(*) as city_count
...> from history
...> group by century
...> order by century desc
...> ;

```

dates	city_count
21-й век	1
20-й век	263
19-й век	189
18-й век	191
17-й век	137
16-й век	79
15-й век	39
14-й век	38
13-й век	27
12-й век	44
11-й век	8
10-й век	6
9-й век	4
5-й век	2
3-й век	1
1-й век	88

Рисунок 7.16 – Подзапросы

```

sqlite> .mode json
sqlite> select kladr_id, city
...> from city
...> where region = 'Самарская'
...> limit 3;
[{"kladr_id":"6300000200000","city":"Жигулевск"},
{"kladr_id":"6300001000000","city":"Кинель"},
{"kladr_id":"6301700100000","city":"Нефтегорск"}]

```

Рисунок 7.17 – Экспорт в CSV

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каково назначение реляционных баз данных и СУБД?

Реляционные базы данных нужны для хранения данных в виде таблицы, а системы управления баз данными нужны для взаимодействия с данными.

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. Зарезервированные слова, которыми обычно выступают операторы, принято писать заглавными буквами.

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

SQLite – это система управления базами данных, отличительной особенностью которой является ее встраиваемость в приложения, в то время как большинство СУБД являются самостоятельными приложениями, взаимодействие с которыми организовано по принципу клиент-сервер. Программа-клиент посылает запрос на языке SQL, а СУБД возвращает результат запроса.

5. Как установить SQLite в Windows и Linux?

В Ubuntu установить sqlite3 можно командой `sudo apt install sqlite3`

Для операционной системы Windows скачивают архив с сайта <https://sqlite.org>, и распаковывают. Далее настраивают путь к каталогу, добавляя адрес каталога к переменной PATH

6. Как создать базу данных SQLite?

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?

`CREATE TABLE` и `DROP TABLE` соответственно

9. Что является первичным ключом в таблице?

Для реляционных баз данных важно, чтобы каждую запись-строку таблицы можно было однозначно идентифицировать. То есть в таблицах не должно быть полностью совпадающих строк. Записи должны отличаться хотя бы по одному полю. Такое поле и называется первичный ключ

10. Как сделать первичный ключ таблицы автоинкрементным?

Для создания инкрементного ключа таблицы следует назначить ограничитель `AUTOINCREMENT`

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

Ограничитель `NOT NULL` используют, чтобы запретить оставление поля пустым. Однако, добавив ограничитель `DEFAULT`, вы сможете не указывать значение. `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. Внешний ключ в одной таблице для другой является первичным. Внешние ключи не обязаны быть уникальными.

```
FOREIGN KEY (theme) REFERENCES sections(_id)
```

FOREIGN KEY является ограничителем, так как не дает нам записать в поле столбца какое-либо иное значение, которое не встречается в качестве первичного ключа в другой таблице.

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>, ...);
```

После INSERT INTO указывается имя таблицы, после в скобках перечисляются столбцы.

После VALUES перечисляются данные, вставляемые в поля столбцов.

14. Как выбрать данные из таблицы SQLite?

С помощью оператора SELECT осуществляется выборочный просмотр данных из таблицы. Например, простейший просмотр всех столбцов будет выглядеть следующим образом:

```
SELECT * FROM <table_name>;
```

15. Как ограничить выборку данных с помощью условия WHERE?

С помощью WHERE определяются строки, которые будут выбраны, обновлены или удалены. После ключевого слова WHERE записывается логическое выражение. Например:

```
sqlite> SELECT * FROM pages
...> WHERE _id == 3;
```

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?

Обновление записей можно выполнить с помощью команды UPDATE.

Синтаксис:

```
UPDATE имя_таблицы  
SET имя_столбца = новое_значение  
WHERE условие;
```

18. Как удалить записи из таблицы SQLite?

Удаление выполняется командой DELETE FROM

```
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.

GROUP BY группирует все записи, в которых встречается одно и то же значение в указанном столбце, в одну строку.

20. Как получить значение агрегатной функции (например: минимум, максимум, количество записей и т. д.) в выборке из таблицы SQLite?

Для этих целей в SQL предусмотрены различные функции агрегирования данных. Наиболее используемые – count(), sum(), avr(), min(), max()

21. Как выполнить объединение нескольких таблиц в операторе SELECT?

В SQL для соединения данных из разных таблиц используется оператор JOIN. В нашем случае запрос будет выглядеть следующим образом:

```
SELECT pages.title,  
       sections.name AS theme  
FROM pages JOIN sections  
ON pages.theme == sections._id;
```

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

Подзапрос позволяет объединять два запроса в один. Шаблон позволяет искать записи, если неизвестно полное имя поля.

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. Таблица виртуальная потому, что на самом деле ее нет в базе данных. В такую таблицу не получится вставить данные, обновить их или удалить. Можно только посмотреть хранящиеся в ней данные, сделать из нее выборку.

24. Какие существуют средства для импорта данных в SQLite?

Команда .import

25. Каково назначение команды .schema ?

Она показывает схему данных всей таблицы.

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

Командами GROUP BY и ORDER BY

27. Каково назначение "табличных выражений" в SQLite?

Это обычный селект, к которому можно для краткости обращаться по имени, как к таблице.

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

Командой .once или output направляя вывод в указанный файл

29. Какие еще форматы для экспорта данных Вам известны?

Markdown, HTML. Также Экспорт таблицы может осуществляться в формат текстовых файлов (*.txt, *.csv), файлов SQL-запросов (*.sql), баз данных SQLite (*.sqlite, *.sqlitedb), баз данных Microsoft Access (*.mdb, *.accdb), баз данных Microsoft SQL Server (*.mdf), таблиц Paradox (*.db) и таблиц dBase (*.dbf).