

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙ-
СКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.20

Тема: «Основы
работы с SQLite3»

Выполнил студент группы

ИВТ-б-о-21-1

Эсеналиев А.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

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

Ход работы:

1. Создал репозиторий в GitHub, дополнил правила в .gitignore для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на компьютер и организовал в соответствии с моделью ветвления git-flow.

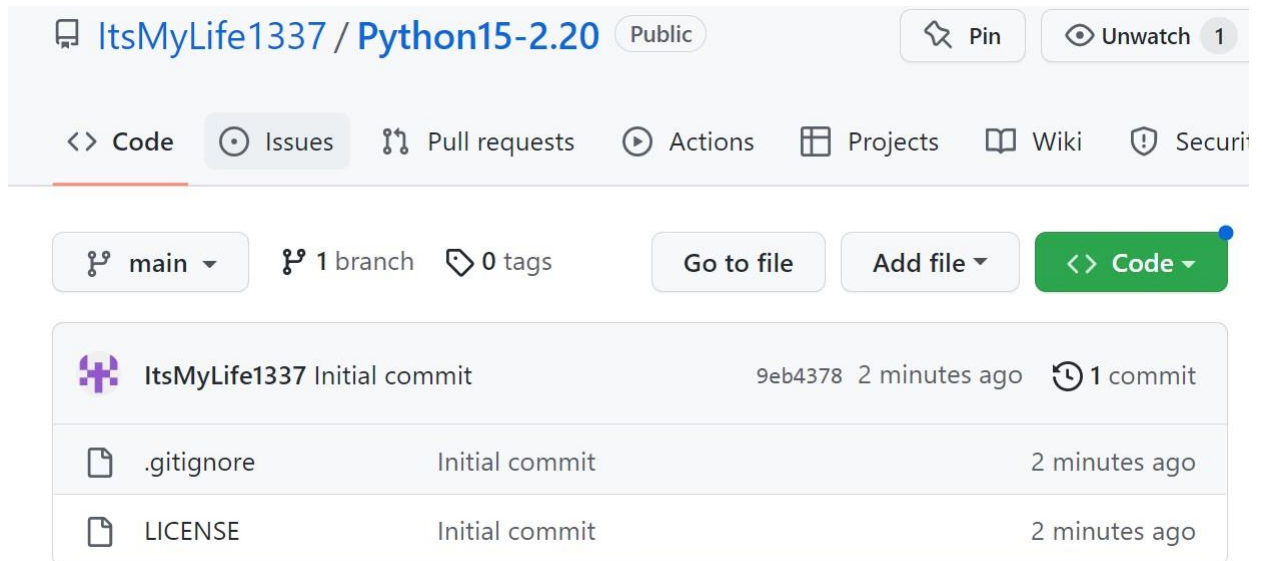


Рисунок 1.1 – Созданный репозиторий

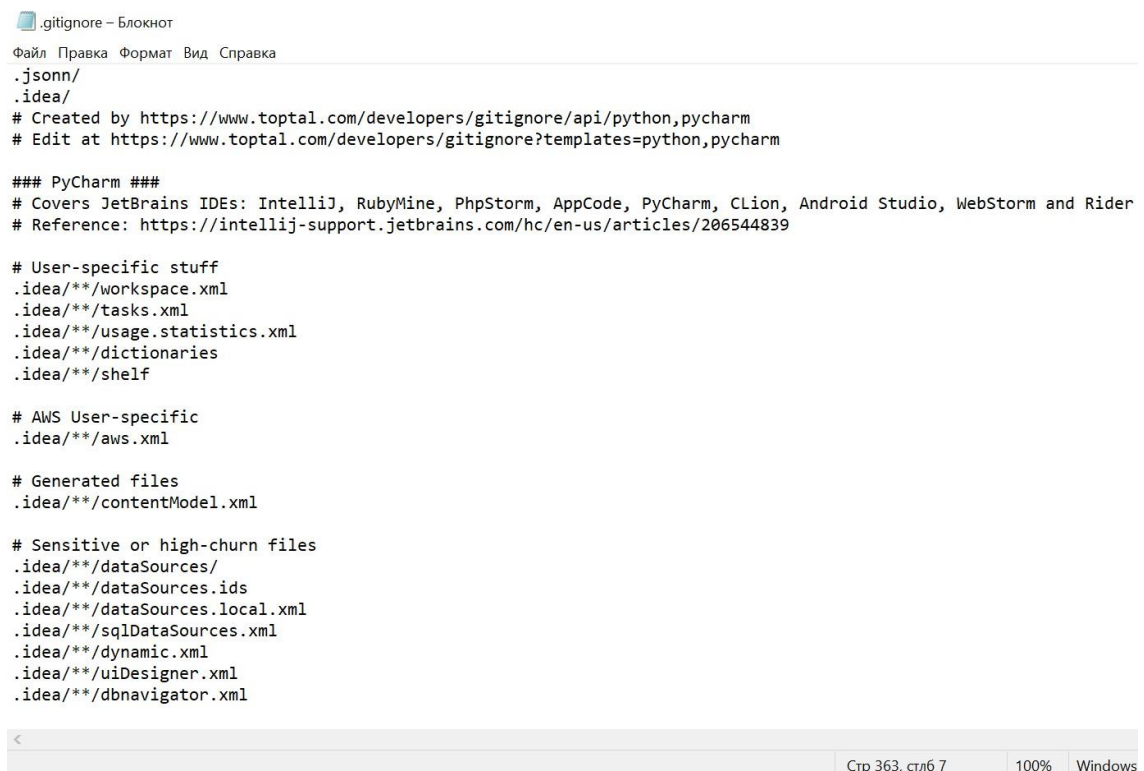


Рисунок 1.2 – Дополнил правила в .gitignore

```

c:\Users\Admin\Desktop\git\Python15-2.20>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Admin/Desktop/git/Python15-2.20/.git/hooks]

c:\Users\Admin\Desktop\git\Python15-2.20>

```

Рисунок 1.3 – Организация репозитория в соответствии с моделью ветвления git-flow

2. Изучил работу примеров лабораторной работы в «Песочнице» и приступил к выполнению заданий.

Задание №1(7). Выполнение команд. Вот что здесь происходит:

```

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);

```

Рисунок 2 – Создание таблицы customer со столбцом (name)

Что вернула команда .schema? Ответ: данная команда показала какие столбцы есть в таблице.

Задание №2(8). Решите задачу: с помощью команды .help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса. Если ее включить, в результатах запроса добавится строчка:

Например: Run Time: real XXX user XXX sys XXX

```
.system CMD ARGS...      Run CMD ARGS... in a system shell
.tables ?TABLE?          List names of tables matching LIKE pattern TA
.testcase NAME            Begin redirecting output to 'testcase-out.txt
.testctrl CMD ...         Run various sqlite3_test_control() operations
.timeout MS               Try opening locked tables for MS milliseconds
.timer on|off             Turn SQL timer on or off
.trace ?OPTIONS?          Output each SQL statement as it is run
.vfsinfo ?AUX?            Information about the top-level VFS
.vfslist                  List all available VFSes
.vfsname ?AUX?            Print the name of the VFS stack
.width NUM1 NUM2 ...      Set minimum column widths for columnar output
sqlite> 
```

Рисунок 3 – С помощью команды .help нашёл команду, которая отвечает за время выполнения запроса

```
sqlite> .timer on
```

Рисунок 4 – Включаем таймер (чтобы увидеть время выполнения запросов)

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

count (*)
1117

Run Time: real 0.000 user 0.000255 sys 0.000000
sqlite> 
```

Рисунок 5 – Вводим необходимый запрос и получаем время его выполнения

Задание №3(9). Решите задачу: загрузите файл city.csv в песочнице. Затем выполните такой запрос: `select max(length(city)) from city;`

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

max(length(city))
25

sqlite> 
```

Рисунок 6 – Вывод запроса

Какое число он вернул? **Ответ: 25**

Задание №4(10). Решите задачу: 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.

```
Last login: Sun Dec 25 14:34:06 2022 from 127.0.0.1
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> .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
  --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.
sqlite> .mode csv
sqlite> .import city.csv city
sqlite> 
```

Рисунок 7 – Добавление данных без использования опции `--csv`

Задание №5(11). 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.

Запрос:

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;
```

Рисунок 8 – Написанный запрос

```
UTC+3 | 101
UTC+4 | 41
UTC+5 | 58
UTC+6 | 6
UTC+7 | 86
UTC+8 | 22
sqlite> █
```

Рисунок 9 – Ответ

Задание №6(12). Решите задачу: напишите в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару. Укажите в ответе названия этих трех городов через запятую в порядке удаления от Самары.

Запрос:

```
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 = '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;
```

Рисунок 10 – Написанный запрос

```
0.00105299999999886 | Заречный
0.0094843000000004 | Каменка
0.01199310000000051 | Елизovo
```

Рисунок 11 – Результат выполнения запроса

Задание №7(13). Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

А теперь выполните этот же запрос, но так, чтобы результат был

- в формате CSV,
- с разделителем «pipe» |

Как выглядит четвертая строка результата?

```
sqlite> select timezone,
...> count (*) city_count
...> from city
...> group by 1
...> order by 2 desc;
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 12 – Результат выполнения запроса в формате csv с заголовками

```
sqlite> .separator |
sqlite> select timezone,
...> count(*) city_count
...> from city
...> group by 1
...> order by 2 desc;
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 13 – Результат выполнения запроса с «pipe» разделителем

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. Выбранный [датасет](#).

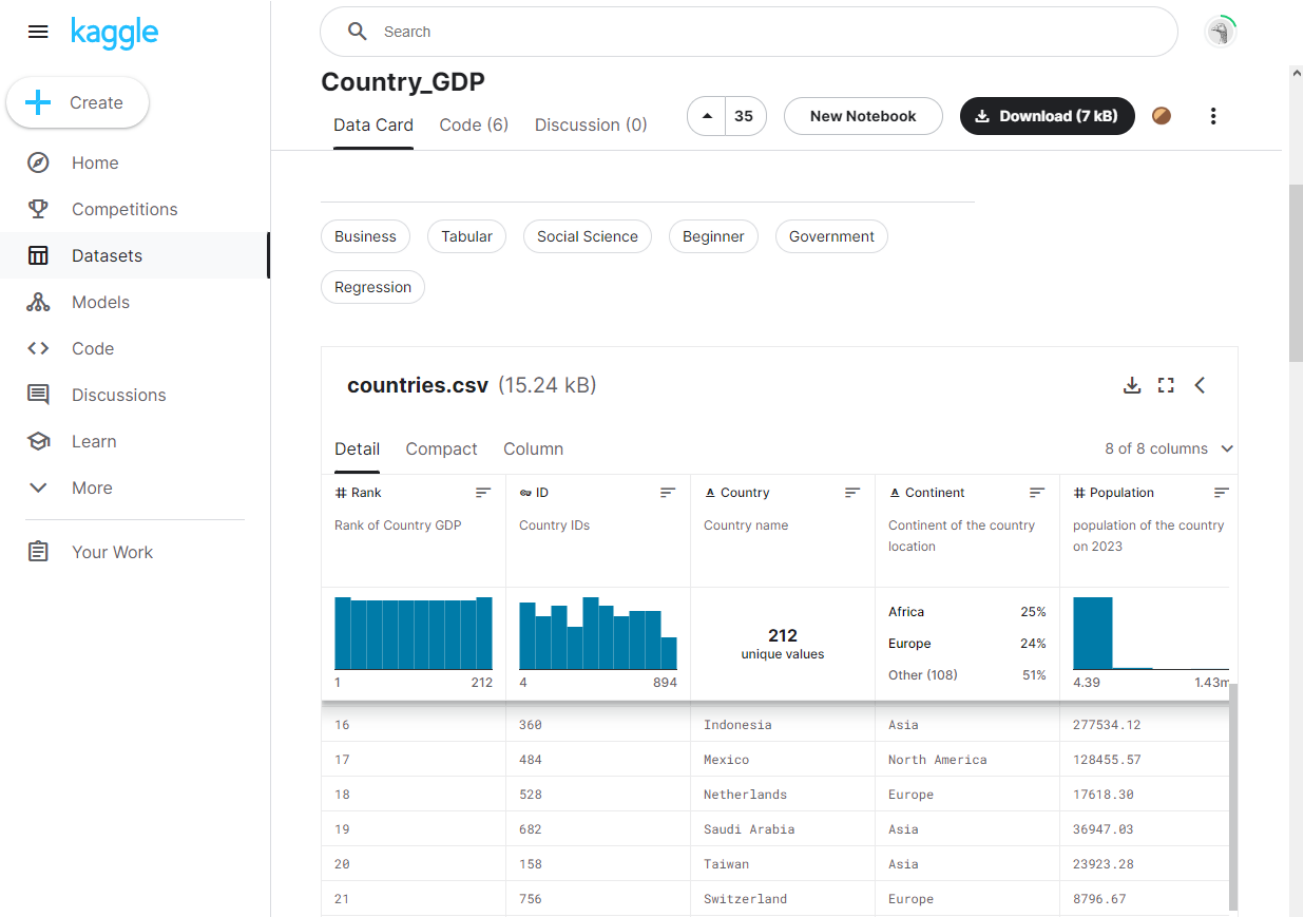


Рисунок 14 – Выбранный датасет с Kaggle

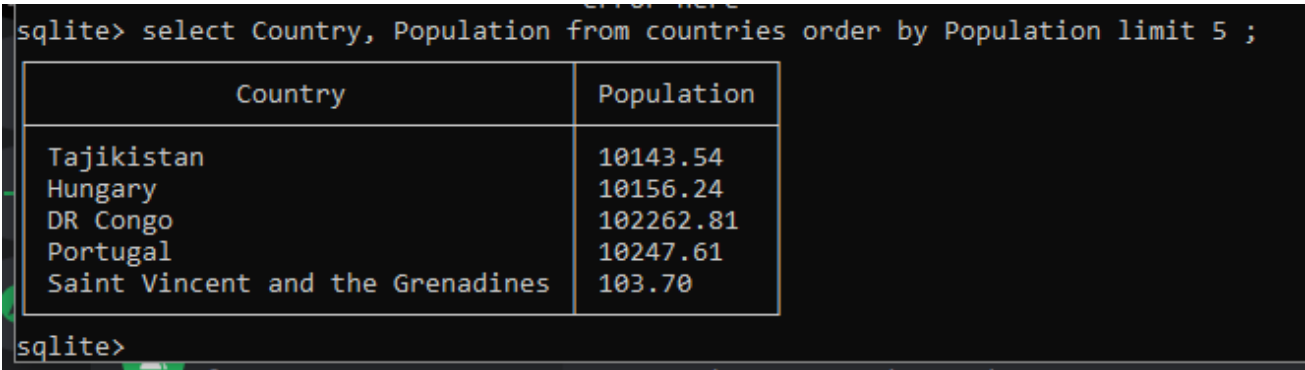


Рисунок 15 – Запрос 1


```
sqlite> select ID, Country from countries where Rank == 16 ;
```

ID	Country
360	Indonesia

Рисунок 16 – Запрос 2

```
sqlite> select Country, Continent from countries where Rank >= 5;
```

Country	Continent
India	Asia
United Kingdom	Europe
France	Europe
Canada	North America
Italy	Europe
Portugal	Europe
Peru	South America
Greece	Europe
Qatar	Asia
Kazakhstan	Europe
Hungary	Europe
Algeria	Africa
Kuwait	Asia
Morocco	Africa
Angola	Africa
Slovakia	Europe
Puerto Rico	North America
Ecuador	South America
Kenya	Africa
Dominican Republic	North America
Ethiopia	Africa
Oman	Asia
Bulgaria	Europe
Guatemala	North America
Luxembourg	Europe
Sri Lanka	Asia
Tanzania	Africa
Turkmenistan	Asia
Uzbekistan	Asia
Ivory Coast	Africa
Ghana	Africa
Panama	North America
Lithuania	Europe
Croatia	Europe
Myanmar	Asia
Azerbaijan	Europe
Serbia	Europe
DR Congo	Africa
Slovenia	Europe
Costa Rica	North America
Uruguay	South America
Belarus	Europe
Jordan	Asia
Uganda	Africa
Venezuela	South America
Cameroon	Africa
Bahrain	Asia
Libya	Africa
Macau	Asia
Paraguay	South America
Bolivia	South America
Latvia	Europe
Estonia	Europe
Nepal	Asia
Zimbabwe	Africa

Рисунок 17 – 3 Запрос

```
sqlite> select Rank, ID from countries where Population >= 100000.00 limit 5;
```

Rank	ID
1	840
2	156
3	392
4	276
5	356

```
sqlite>
```

Рисунок 18 – Запрос 4

```
sqlite> select Rank, Country, Population from countries where Rank > 10 and Continent == 'Europe';
```

Rank	Country	Population
4	Germany	83294.63
6	United Kingdom	67736.80
7	France	64756.58
9	Italy	58870.76
14	Russia	144444.36
15	Spain	47519.63
18	Netherlands	17618.30
21	Switzerland	8796.67
22	Poland	41026.07
23	Turkey	85816.20
24	Sweden	10612.09
25	Belgium	11686.14
28	Ireland	5056.94
30	Norway	5474.36
32	Austria	8958.96
41	Denmark	5910.91
45	Czech Republic	10495.30
46	Finland	5545.48
47	Romania	19892.81
50	Portugal	10247.61
52	Greece	10341.28
54	Kazakhstan	19606.63
55	Hungary	10156.24
60	Slovakia	5795.20
67	Bulgaria	6687.72
69	Luxembourg	654.77
77	Lithuania	2718.35
78	Croatia	4008.62
80	Azerbaijan	10412.65
81	Serbia	7149.08
83	Slovenia	2119.68
86	Belarus	9498.24
96	Latvia	1830.21
97	Estonia	1322.77
105	Cyprus	1260.14
107	Iceland	375.32
111	Bosnia and Herzegovina	3210.85
112	Georgia	3728.28
122	Albania	2832.44
123	Malta	535.06
129	Armenia	2777.97
131	North Macedonia	2085.68
135	Moldova	3435.93
147	Montenegro	626.49
156	Andorra	80.09
171	San Marino	33.64
198	Kosovo	1663.59
200	Liechtenstein	39.58
201	Monaco	36.30
212	Ukraine	36744.63

Рисунок 19 – Запрос 5

```
sqlite> select Rank, Country, Population, IMF_GDP from countries where Continent == 'Asia' order by Population desc limit 5;
```

Rank	Country	Population	IMF_GDP
35	Vietnam	98858.95	462644000000.00
33	United Arab Emirates	9516.87	506234000000.00
31	Israel	9174.52	547765000000.00
13	Iran	89172.77	1782602000000.00
159	Bhutan	787.42	2881000000.00

Рисунок 20 – Запрос 6













 request1	25.12.2022 16:40	Microsoft Excel Com...	1 КБ
 request1	25.12.2022 18:18	JSON File	1 КБ
 request2	25.12.2022 18:00	Microsoft Excel Com...	1 КБ
 request2	25.12.2022 18:18	JSON File	1 КБ
 request3	25.12.2022 17:59	Microsoft Excel Com...	1 КБ
 request3	25.12.2022 18:18	JSON File	1 КБ
 request4	25.12.2022 18:01	Microsoft Excel Com...	1 КБ
 request4	25.12.2022 18:18	JSON File	1 КБ
 request5	25.12.2022 18:05	Microsoft Excel Com...	1 КБ
 request5	25.12.2022 18:18	JSON File	1 КБ
 request6	25.12.2022 18:08	Microsoft Excel Com...	2 КБ
 request6	25.12.2022 18:16	JSON File	2 КБ

Рисунок 21 – Все 6 результатов выгруженные в формате .JSON и .csv

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

Ответы на 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 результатзапроса.

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 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 поимени 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`