

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное
учреждение высшего образования**
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра инфокоммуникаций
Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.20
Дисциплина: «Программирование на Python»
Тема: «Основы работы с SQLite3»

Выполнила: студентка 1 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2023

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

Практическая часть:


1. Создала общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *


 DinaDichenko ▾

 /


Lab2_20 ✓

Great repository names are short and memorable. Need inspiration? How about [miniature-octo-funicular?](#)

Description (optional)

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

 You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Выполнила клонирование созданного репозитория на рабочий компьютер.

```
C:\Users\super\OneDrive\Рабочий стол\ВУЗЬ>git clone https://github.com/DinaDichenko/Lab2_20.git
Cloning into 'Lab2_20'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\super\OneDrive\Рабочий стол\ВУЗЬ>
```

Рисунок 2. Клонирование репозитория

3. Дополнила файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.

```
.jsonn/
.idea/
# Created by https://www.toptal.com/developers/gitignore/api/python,pycharm
# Edit at https://www.toptal.com/developers/gitignore?templates=python,pycharm

### PyCharm ###
# Covers JetBrains IDEs: IntelliJ, RubyMine, PhpStorm, AppCode, PyCharm, CLion, Android Studio, WebStorm and Rider
# Reference: https://intellij-support.jetbrains.com/hc/en-us/articles/206544839

# User-specific stuff
.idea/**/workspace.xml
.idea/**/tasks.xml
.idea/**/usage.statistics.xml
.idea/**/dictionaries
.idea/**/shelf

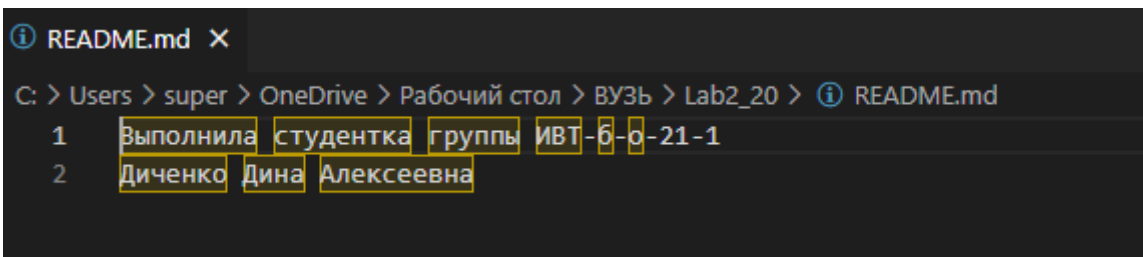
# AWS User-specific
.idea/**/aws.xml

# Generated files
.idea/**/contentModel.xml

# Sensitive or high-churn files
.idea/**/dataSources/
.idea/**/dataSources.ids
.idea/**/dataSources.local.xml
.idea/**/sqlDataSources.xml
.idea/**/dynamic.xml
```

Рисунок 3. Изменение файла .gitignore

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



```
1 Выполнила студентка группы ИВТ-6-0-21-1
2 Диченко Дина Алексеевна
```

Рисунок 4. Изменение файла README.md

5. Добавила файл README и зафиксировала сделанные изменения.

```

C:\Users\super\OneDrive\Рабочий стол\ВУЗб\Lab2_20>git commit -m "gitignore&readme"
[main 565c90c] gitignore&readme
 2 files changed, 238 insertions(+), 2 deletions(-)
 create mode 100644 README.md

C:\Users\super\OneDrive\Рабочий стол\ВУЗб\Lab2_20>git push
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 2.91 KiB | 745.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DinaDichenko/Lab2_20.git
 a3f5d81..565c90c  main -> main

```

Рисунок 5. Фиксация изменений

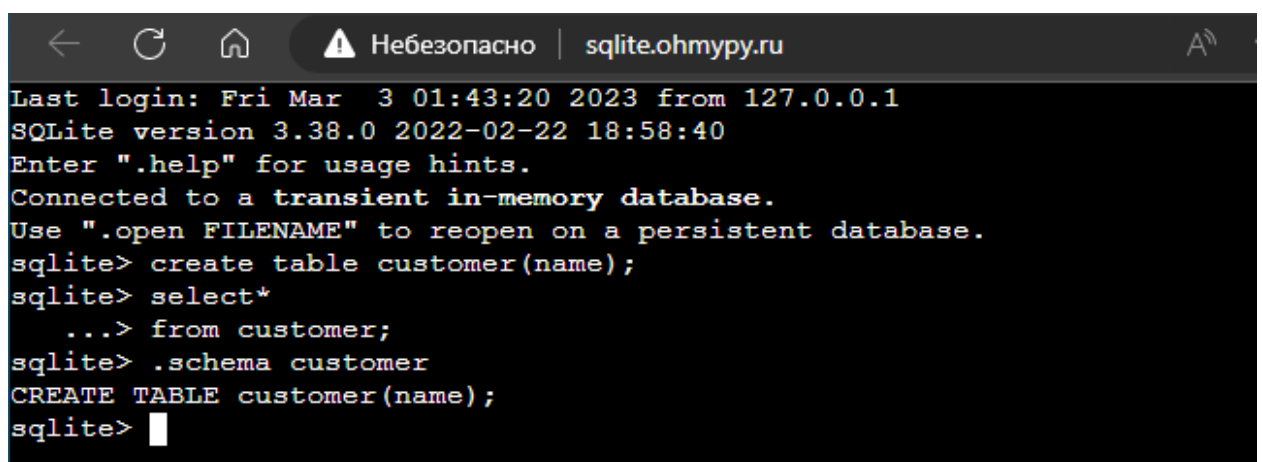
6. Решила задачу: выполните в песочнице команды:

create table customer(name);

select *

from customer;

.schema customer



```

← ↻ 🏠 ⚠ Небезопасно | sqlite.ohmypy.ru
Last login: Fri Mar 3 01:43:20 2023 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> create table customer(name);
sqlite> select*
...> from customer;
sqlite> .schema customer
CREATE TABLE customer(name);
sqlite>

```

Рисунок 6. Решение задачи 1

Что вернула .schema?

Ответ: данная команда показала какие столбцы есть в таблице.

7. Решила задачу: с помощью команды .help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса. Если ее включить, в результатах запроса добавится строка:

Run Time: real XXX user XXX sys XXX

Например:

sqlite> .SOMETHING on

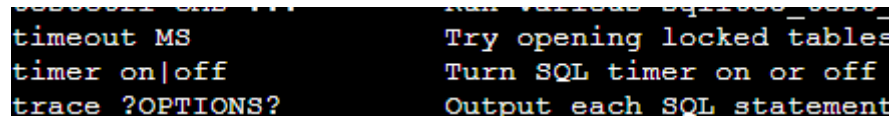
```
sqlite> select count(*) from city;
```

```
1117
```

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

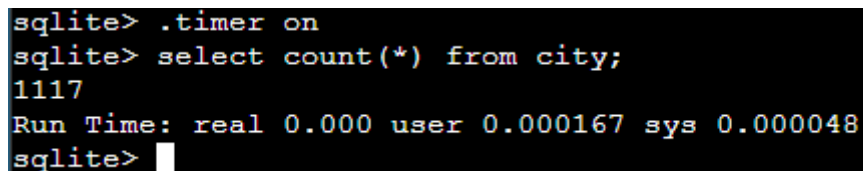
Какая команда должна быть вместо SOMETHING?

Ответ: .timer on|off



```
timeout MS          Try opening locked tables
timer on|off        Turn SQL timer on or off
trace ?OPTIONS?     Output each SQL statement
```

Рисунок 7. Поиск нужной команды



```
sqlite> .timer on
sqlite> select count(*) from city;
1117
Run Time: real 0.000 user 0.000167 sys 0.000048
sqlite> 
```

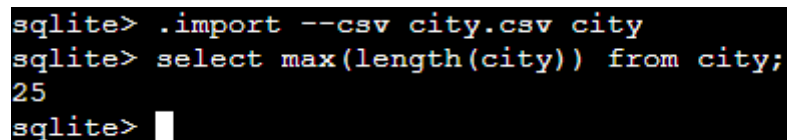
Рисунок 8. Решение задачи 2

8. Решите задачу: загрузите файл city.csv в песочнице:

```
.import --csv city.csv city
```

Затем выполните такой запрос:

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



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

Рисунок 8. Решение задачи 3

Какое число он вернул?

Ответ: 25.

9. Решила 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. Вам поможет команда .help import . Всего должно получиться две команды:

```
do_something
```

```
.import city.csv city
```

```

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.

```

Рисунок 9. Поиск нужной команды

Какая команда должна быть вместо do_something?

Ответ: .mode

```

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

```

Рисунок 10. Решение задачи 4

10. Решила 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:

timezone	city_count
UTC+3	xxx
UTC+4	xx
UTC+5	xx
UTC+6	x
UTC+7	xx
UTC+8	xx

```

sqlite> .mode box
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

```

sqlite>

```

Рисунок 11. Решение задачи 5

Укажите в ответе значение city_count для timezone = UTC+5 .

Ответ: 58.

11. Решила 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.

```
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_lo
u from city)
...> select sqrt((power((geo_las - geo_lam),2) + power((geo_los - geo_lou),2))) As city from (geo
_las, geo_los, geo_lam, geo_lou)
...> where city != 'Самара'
...> order by distance limit 3;
Parse error: no such column: distance
am, geo_lou) where city != 'Самара' order by distance limit 3;
error here ---^
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_lo
u 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 limit 3;
0.001052999999999886,"Заречный"
0.0094843000000004,"Каменка"
0.01199310000000051,"Елизово"
sqlite>
```

Рисунок 12. Решение задачи 6

12. Решила задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

```
sqlite> .open city-1.db
sqlite> .mode box
sqlite> select 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 13. Решение задачи 7

А теперь выполните этот же запрос, но так, чтобы результат был в формате CSV, с заголовками, с разделителем «pipe» |

```
sqlite> select timezone, count(*) as city_count
...> from city
...> group by timezone
...> order by city_count 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 14. Решение задачи 7 с «pipe» разделителем

13. Выполнила 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.

Датасет: [Dog breeds details | Kaggle](https://www.kaggle.com/datasets/warcodeer/dog-breeds-details).

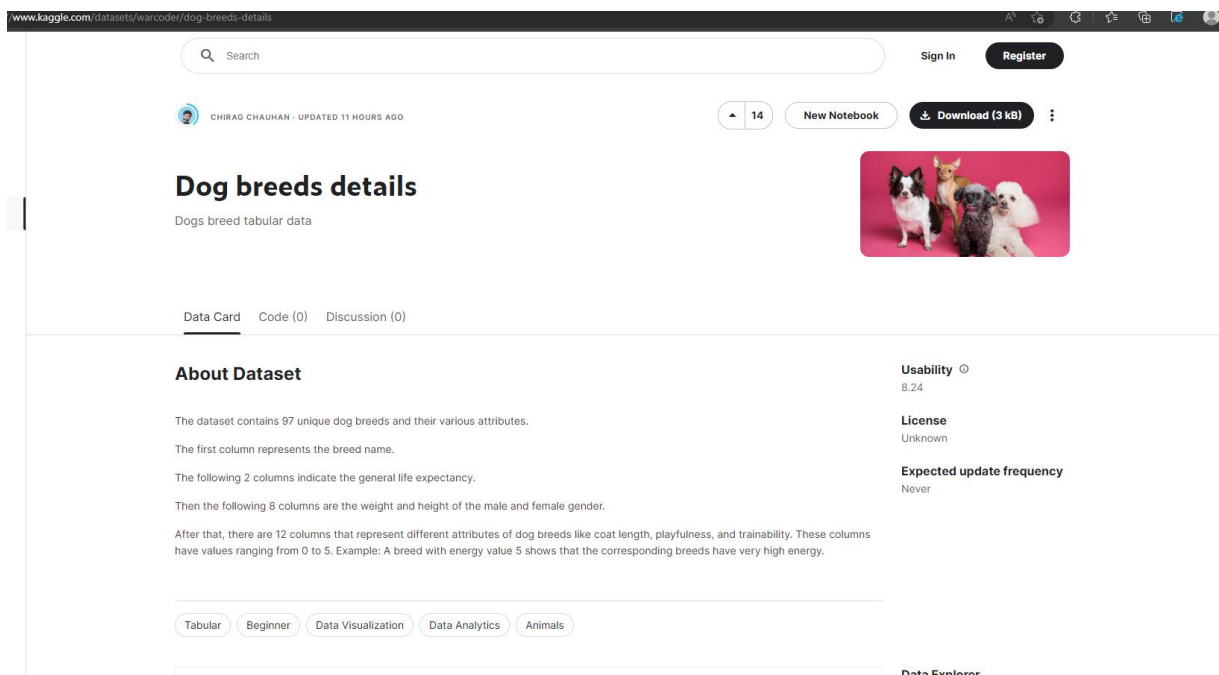
The image shows a screenshot of a Kaggle dataset page. At the top, there's a search bar and navigation links like 'Sign In' and 'Register'. Below that, the dataset title 'Dog breeds details' is prominently displayed, along with a small image of various dog breeds. The page includes tabs for 'Data Card', 'Code (0)', and 'Discussion (0)'. The 'About Dataset' section provides details about the data, stating it contains 97 unique dog breeds and lists various attributes. On the right side, there are sections for 'Usability' (8.24), 'License' (Unknown), and 'Expected update frequency' (Never). At the bottom, there are tags for 'Tabular', 'Beginner', 'Data Visualization', 'Data Analytics', and 'Animals', followed by a 'Data Explorer' button.

Рисунок 15. Выбранный датасет с сайта Kaggle


```
sqlite> select Name, min_life_expectancy from dog where min_life_expectancy='10' order by Name;
```

Name	min_life_expectancy
Akita	10
Alaskan Malamute	10
American Bulldog	10
American Water Spaniel	10
Beagle	10
Beauceron	10
Bloodhound	10
Boxer	10
Bracco Italiano	10
Cocker Spaniel	10
Doberman Pinscher	10
French Bulldog	10
Golden Retriever	10
Great Pyrenees	10
Greyhound	10
Hovawart	10
Japanese Chin	10
Komondor	10
Labrador Retriever	10
Otterhound	10
Poodle (Miniature)	10
Rhodesian Ridgeback	10
Shih Tzu	10
Tibetan Mastiff	10

Рисунок 16. Первый запрос

```
sqlite> select Name, shedding, grooming from dog order by shedding limit 20;
```

Name	shedding	grooming
Plott Hound	0	0
Poodle (Miniature)	1	4
Shih Tzu	1	4
Maltese	1	4
Bichon Frise	1	5
Komondor	1	4
Afghan Hound	1	4
Bolognese	1	3
American Hairless Terrier	1	1
American Water Spaniel	1	3
Barbet	1	3
Bergamasco Sheepdog	1	1
Pumi	1	2
Xoloitzcuintli	1	1
Yorkshire Terrier	1	5
Dachshund	2	2
Boxer	2	2
Pomeranian	2	3
Greyhound	2	1
Basset Hound	2	3

Рисунок 17. Второй запрос

```
sqlite> select Name, max_height_male, max_height_female, max_weight_male, max_weight_female from dog order by Name limit 20;
```

Name	max_height_male	max_height_female	max_weight_male	max_weight_female
Afghan Hound	27	27	60	60
Akita	28	28	130	100
Alaskan Klee Kai	28	28	130	100
Alaskan Malamute	25	25	85	75
American Bulldog	28	24	100	80
American Eskimo Dog	26	26	65	65
American Hairless Terrier	16	16	16	16
American Leopard Hound	27	27	70	70
American Staffordshire Terrier	19	19	70	55
American Water Spaniel	18	18	45	40
Anatolian Shepherd Dog	29	29	150	120
Australian Cattle Dog	20	20	50	50
Australian Shepherd	23	23	65	55
Barbet	24.5	24.5	65	65
Basenji	17	17	24	22
Basset Hound	15	14	65	65
Beagle	16	15	20	30
Beauceron	27.5	27.5	110	110
Belgian Malinois	26	26	80	60
Bergamasco Sheepdog	23.5	23.5	84	71

Рисунок 18. Третий запрос

```
sqlite> select Name, good_with_children, good_with_strangers, playfulness from dog order by playfulness;
```

Name	good_with_children	good_with_strangers	playfulness
American Bulldog	3	0	0
Plott Hound	0	0	0
Alaskan Klee Kai	0	0	0
Bullmastiff	3	3	3
Shih Tzu	5	3	3
Pomeranian	3	3	3
Greyhound	3	3	3
Chow Chow	3	2	3
Chinese Shar-Pei	3	3	3
Cocker Spaniel	5	4	3
Great Pyrenees	3	3	3
Newfoundland	5	5	3
Basset Hound	5	3	3
Maltese	3	3	3
Akita	3	2	3
Bloodhound	3	3	3
Komondor	3	3	3
Otterhound	3	4	3

Рисунок 19. Четвертый запрос

```
sqlite> select Name, trainability, energy
...> from dog
...> where Name glob 'A*';
```

Name	trainability	energy
American Bulldog	4	0
Akita	3	4
Afghan Hound	1	4
American Eskimo Dog	4	4
American Hairless Terrier	5	3
American Leopard Hound	3	4
American Water Spaniel	5	3
Anatolian Shepherd Dog	2	3
Alaskan Klee Kai	3	4
Alaskan Malamute	5	4
Australian Cattle Dog	4	5
American Staffordshire Terrier	3	3
Australian Shepherd	5	5

Рисунок 20. Пятый запрос

Контрольные вопросы:

1. Каково назначение реляционных баз данных и СУБД?

Теперь вернемся к вопросу о том, что такое реляционная базы данных (РБД). 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). Возможно как и в Linux работает вызов утилиты по ее адресу. Android же имеет уже встроенную библиотеку SQLite.

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?

Чтобы исключить возможность ввода одинаковых идентификаторов, столбец ID назначают первичным ключом.

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.

14. Как выбрать данные из таблицы SQLite?

С помощью оператора SELECT.

15. Как ограничить выборку данных с помощью условия WHERE?

Такая команда отображает значения всех столбцов и строк заданной таблицы. На выборку всех столбцов указывает звездочка после слова SELECT. А все строки будут выбраны потому, что после имени таблицы нет оператора WHERE языка SQL. 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.

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