

**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 «Основы программной инженерии»

Выполнил студент группы

ПИЖ-б-о-21-1

Зиберов Александр

« » мая 2023 г.

Подпись студента _____

Работа защищена

« » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь, 2023

Цель работы:

Исследовать базовые возможности системы управления базами данных SQLite3.

Выполнение работы:

Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT, рисунок 1.

Ссылка: <https://github.com/afk552/lab2.20>

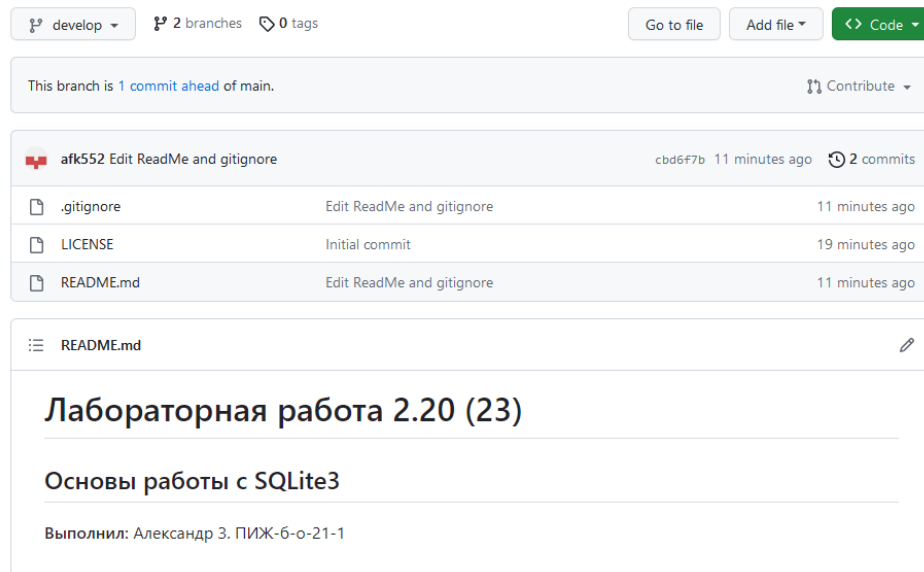


Рисунок 1 – Удаленный репозиторий на GitHub

Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm, рисунок 2.

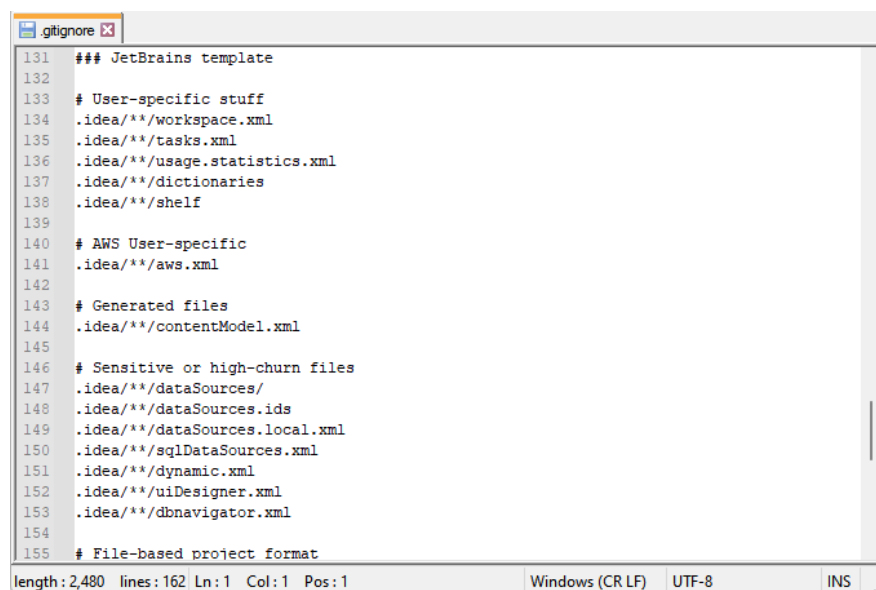


Рисунок 2 – Окно блокнота

Организируйте свой репозиторий в соответствии с моделью ветвления git-flow, рисунок 3.

```
C:\git\lab2.20>git checkout -b develop
Switched to a new branch 'develop'

C:\git\lab2.20>git branch
* develop
  main

C:\git\lab2.20>_
```

Рисунок 3 – Окно командной строки

Примеры (общие задания)

7. Решите задачу: выполните в песочнице команды. Что вернула команда .schema?

```
SQLite version 3.41.2 2023-03-22 11:56:21
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 4 – Выполнение примера 7

Команда .schema показывает список и структуру всех таблиц в базе.

8. Решите задачу: с помощью команды .help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса. Какая команда должна быть вместо SOMETHING?

(Команда .timer)

```
E:\2.20_try>sqlite3 city-1.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .timer on
sqlite> select count(*) from city;
1117
Run Time: real 0.000 user 0.000000 sys 0.000000
sqlite>
```

Рисунок 5 – Выполнение примера 8

9. Решите задачу: загрузите файл city.csv в песочнице, затем выполните такой запрос: `select max(length(city)) from city;` Какое число он вернул?

```
E:\2.20_try>sqlite3 ex9.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .import --csv city.csv city
sqlite> select max(length(city)) from city;
25
sqlite>
```

Рисунок 6 – Выполнение примера 9

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. Всего должно получиться две команды:

```
I:\2.20_try>sqlite3 ex9.db
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
sqlite> .mode csv
sqlite> .import city.csv city
sqlite> _
```

Рисунок 7 – Выполнение примера 10

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:

```
ex11.sql
1 .mode box
2 .import --csv city.csv city
3 SELECT timezone, count(*) AS city_count
4 FROM city WHERE federal_district = 'Сибирский' or federal_district = 'Приволжский'
5 GROUP BY timezone
6 ORDER BY timezone ASC;
```

Рисунок 8 – Текст запроса примера 11

```
I:\2.20_try>sqlite3
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .read ex11.sql
```

timezone	city_count
UTC+3	101
UTC+4	41
UTC+5	58
UTC+6	6
UTC+7	86
UTC+8	22

```
sqlite>
```

Рисунок 9 – Выполнение примера 11

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.

```
ex12.sql
1 .mode box
2 .import --csv city.csv city
3 with dist as(SELECT address, ((53.1950306 - geo_lat)*(53.1950306 - geo_lat))+((50.1069518-geo_lon)*(50.1069518-geo_lon)) as distance from city)
4 SELECT address, distance
5 FROM dist
6 WHERE address <> 'г Самара'
7 ORDER BY distance ASC
8 LIMIT 3;
```

Рисунок 10 – Текст запроса примера 12

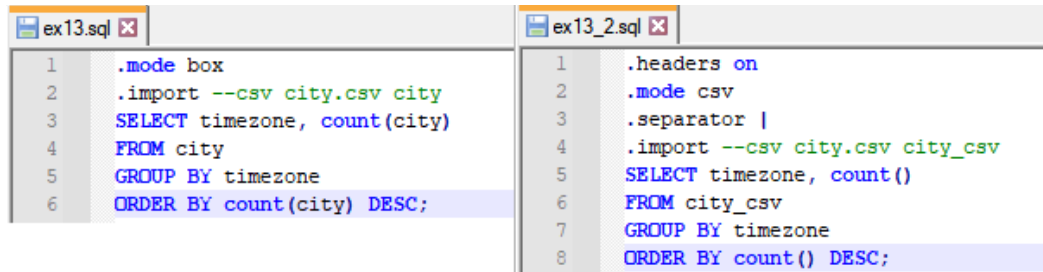
```
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .read ex12.sql
```

address	distance
Самарская обл, г Новокуйбышевск	0.0344833790157681
Самарская обл, г Чапаевск	0.128213124744169
Самарская обл, г Кинель	0.278853932906283

```
sqlite>
```

Рисунок 11 – Текст запроса примера 12

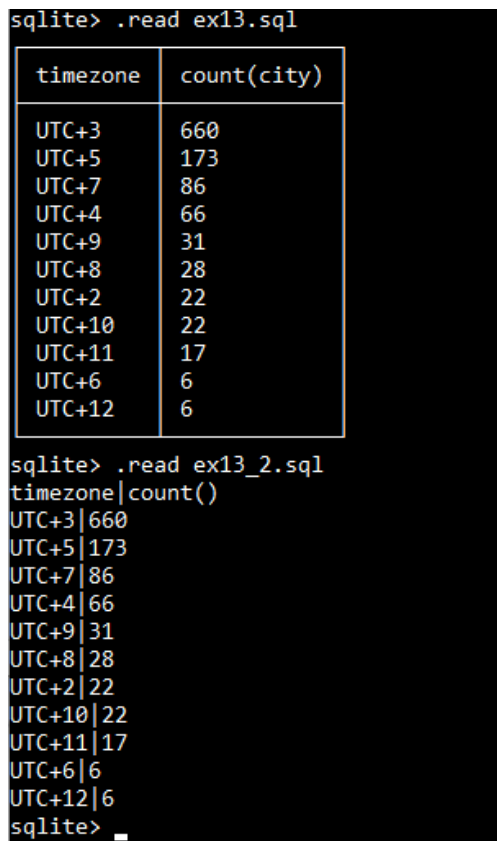
13. Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию. А теперь выполните этот же запрос, но так, чтобы результат был в формате CSV, с заголовками, с разделителем «pipe»



```
ex13.sql
1 .mode box
2 .import --csv city.csv city
3 SELECT timezone, count(city)
4 FROM city
5 GROUP BY timezone
6 ORDER BY count(city) DESC;

ex13_2.sql
1 .headers on
2 .mode csv
3 .separator |
4 .import --csv city.csv city_csv
5 SELECT timezone, count()
6 FROM city_csv
7 GROUP BY timezone
8 ORDER BY count() DESC;
```

Рисунок 12 – Текст запроса примера 13



```
sqlite> .read ex13.sql

  timezone  count(city)
  -----
  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> .read ex13_2.sql
timezone|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 – Текст запроса примера 13

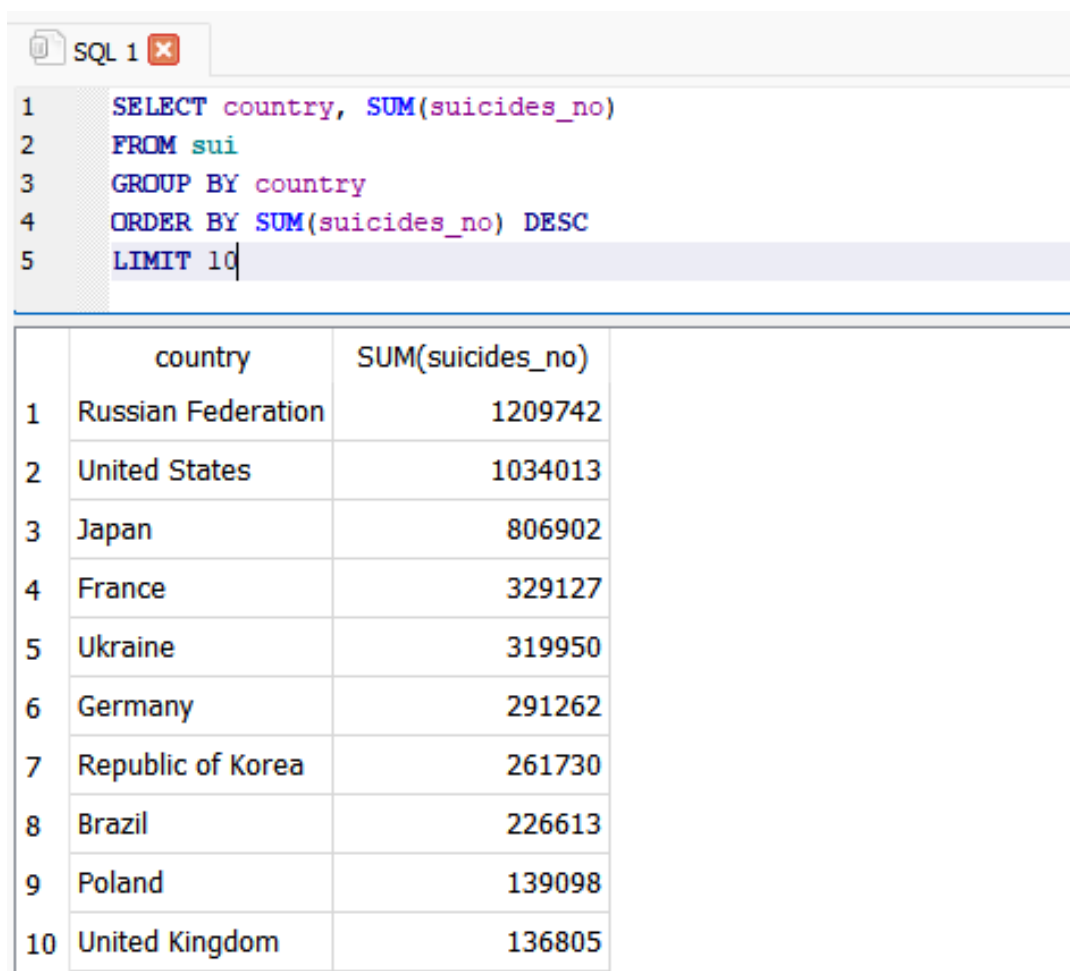
Индивидуальное задание.

Выполните индивидуальное задание. Каждый запрос к базе данных сохраните в файл с расширением sql.

Набор данных содержит показатели числа самоубийств с 1985 по 2016 год.

Country – страна, year – год, sex – пол, age – возрастная группа, suicide_no – количество самоубийств, population – население (годовое), suicides_100k_pop – количество самоубийств на 100 тыс. населения, country-year – пара страна/год, generation – поколение.

Запрос 1. Вывести топ 10-стран по количеству самоубийств, отсортировав по их убыванию.

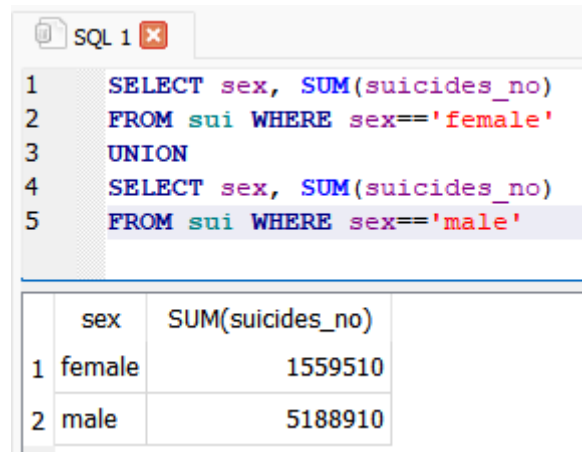


```
SQL 1 X
1 SELECT country, SUM(suicides_no)
2 FROM sui
3 GROUP BY country
4 ORDER BY SUM(suicides_no) DESC
5 LIMIT 10
```

	country	SUM(suicides_no)
1	Russian Federation	1209742
2	United States	1034013
3	Japan	806902
4	France	329127
5	Ukraine	319950
6	Germany	291262
7	Republic of Korea	261730
8	Brazil	226613
9	Poland	139098
10	United Kingdom	136805

Рисунок 14 – Запрос 1

Запрос 2. Вывести количество самоубийств, совершенные мужчинами и женщинами за весь период выборки данных (1985-2016).

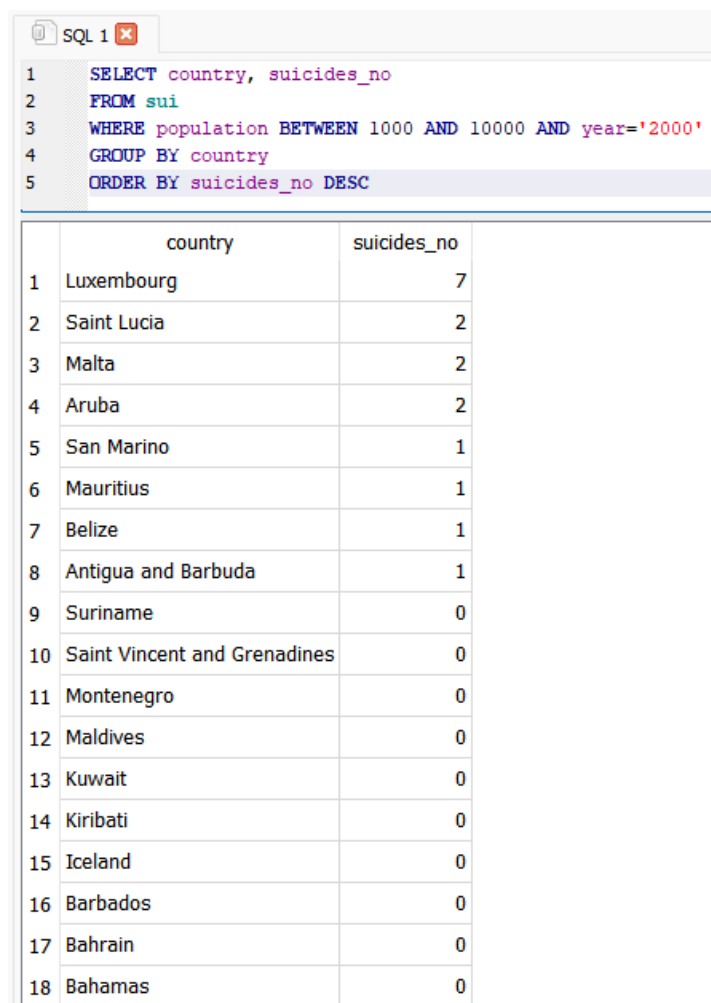


```
1 SELECT sex, SUM(suicides_no)
2 FROM sui WHERE sex=='female'
3 UNION
4 SELECT sex, SUM(suicides_no)
5 FROM sui WHERE sex=='male'
```

	sex	SUM(suicides_no)
1	female	1559510
2	male	5188910

Рисунок 15 – Запрос 2

Запрос 3. Вывести количество самоубийств, совершенных в 2000 году в странах с населением в диапазоне от 1000 до 10000.



```
1 SELECT country, suicides_no
2 FROM sui
3 WHERE population BETWEEN 1000 AND 10000 AND year='2000'
4 GROUP BY country
5 ORDER BY suicides_no DESC
```

	country	suicides_no
1	Luxembourg	7
2	Saint Lucia	2
3	Malta	2
4	Aruba	2
5	San Marino	1
6	Mauritius	1
7	Belize	1
8	Antigua and Barbuda	1
9	Suriname	0
10	Saint Vincent and Grenadines	0
11	Montenegro	0
12	Maldives	0
13	Kuwait	0
14	Kiribati	0
15	Iceland	0
16	Barbados	0
17	Bahrain	0
18	Bahamas	0

Рисунок 16 – Запрос 3

Запрос 4. Вывести информацию о количестве населения и соотношения числа самоубийств на 100 тыс. населения за 2013 год, ограничить вывод запроса до 10 позиций.

SQL 1			
1	SELECT country, population, suicides_100k_pop		
2	FROM sui		
3	WHERE year='2013'		
4	GROUP BY country		
5	ORDER BY suicides_100k_pop DESC		
6	LIMIT 10		

	country	population	suicides_100k_pop
1	Republic of Korea	806960	147.59
2	Slovenia	57600	98.96
3	Lithuania	405191	94.28
4	Austria	255540	86.48
5	Hungary	234783	85.61
6	Serbia	222998	82.96
7	Uruguay	79099	79.65
8	Belarus	169369	73.8
9	Cuba	293384	73.62
10	Estonia	32101	71.65

Рисунок 17 – Запрос 4

Запрос 5. Подсчитать и вывести количество самоубийств, совершенных мужчинами и женщинами в России в 2010 году.

SQL 1		
1	SELECT sex, SUM(suicides_no)	
2	FROM sui WHERE sex='female' and year='2010' and country='Russian Federation'	
3	UNION	
4	SELECT sex, SUM(suicides_no)	
5	FROM sui WHERE sex='male' and year='2010' and country='Russian Federation'	

	sex	SUM(suicides_no)
1	female	5795
2	male	27561

Рисунок 18 – Запрос 5

Вывод: Были изучены базовые возможности системы управления базами данных SQLite3.

Контрольные вопросы:

1. Каково назначение реляционных баз данных и СУБД?

Хранение информации в базах данных дает преимущество не только с точки зрения обеспечения к ним быстрого доступа множества процессов. Базы данных, особенно реляционные, позволяют структурировать данные, манипулирования ими и легко наращивать объем.

Можно сказать, что в одной таблице содержатся ассоциированные данные, а в разных таблицах одной БД находятся связанные данные.

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.

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?

В Ubuntu установить `sqlite3` можно командой `sudo apt install sqlite3`. В этом случае утилита вызывается командой `sqlite3`. Также можно скачать с сайта <https://sqlite.org> архив с последней версией библиотеки, распаковать и вызвать в терминале утилиту.

Для операционной системы Windows скачивают свой архив (`sqlitetools-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?

```
sqlite3 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`.

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. PRIMARY KEY – ограничитель, который заставляет СУБД проверять уникальность значения данного поля у каждой добавляемой записи.

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.

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

Однако в SQLite поддержка внешнего ключа по умолчанию отключена. Поэтому, даже назначив столбец внешним ключом, вы сможете записывать в его поля любые значения. 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.

После INSERT INTO указывается имя таблицы, после в скобках перечисляются столбцы. После слова VALUES перечисляются данные, вставляемые в поля столбцов.

14. Как выбрать данные из таблицы SQLite?

С помощью оператора SELECT осуществляется выборочный просмотр данных из таблицы. В простейшем случае оператор имеет следующий синтаксис, где вместо <table_name> указывается имя 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.

После ключевого слова WHERE записывается логическое выражение, которое может быть как простым (содержащим операторы = или ==, >, <, >=, <=, !=, BETWEEN), так и сложным (AND, OR, NOT, IN, NOT IN).

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 имя_таблицы  
SET имя_столбца = новое_значение  
WHERE условие;
```

18. Как удалить записи из таблицы SQLite?

```
DELETE FROM имя_таблицы WHERE условие;
```

Без WHERE будут удалены все строки, однако сама таблица останется.

19. Как сгруппировать данные из выборки из таблицы SQLite?

```
SELECT theme FROM pages  
GROUP BY theme;
```

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

Вывод количества столбцов таблицы:

```
sqlite> SELECT count() FROM pages;
```

Поиск максимального ID:

```
sqlite> SELECT max(_id) FROM pages;
```

21. Как выполнить объединение нескольких таблиц в операторе SELECT?

```
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 CREATE VIEW, 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?

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

Команда `.import` автоматически создала таблицу `city` со всеми столбцами из `city.csv` и загрузила данные из файла.

25. Каково назначение команды `.schema` ?

Она показывает схему данных всей таблицы.

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?

Упрощение `select`.

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

```
.mode csv  
.mode json
```

29. Какие еще форматы для экспорта данных Вам известны?

XML.