

**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

Богдан А.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

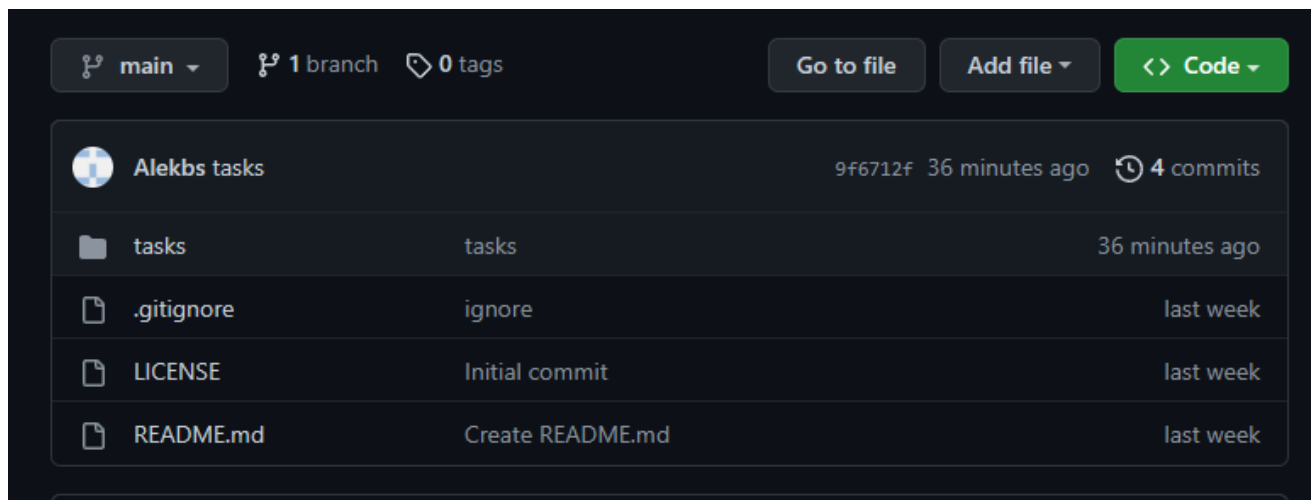
Ставрополь 2023

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

Ход работы:

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



```
.gitignore – Блокнот
Файл Правка Формат Вид Справка
.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
.idea/**/uiDesigner.xml
.idea/**/dbnavigator.xml
```

Рисунок 1.2 – Дополнил правила в .gitignore

```

D:\BY3\python\Lab-2.20>git branch develop

D:\BY3\python\Lab-2.20>git checkout develop
Switched to branch 'develop'

D:\BY3\python\Lab-2.20>git flow init

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

Which branch should be used for integration of the "next release"?
- develop
Branch name for "next release" development: [develop] develop

How to name your supporting branch prefixes?
Feature branches? [feature/] feat
Bugfix branches? [bugfix/] bug
Release branches? [release/] rel
Hotfix branches? [hotfix/] hot
Support branches? [support/] ssup
Version tag prefix? [] pre
Hooks and filters directory? [D:/BY3/python/Lab-2.20/.git/hooks] hook

D:\BY3\python\Lab-2.20>_

```

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

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

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

```

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

```

Рисунок 2 – Создание таблицы customer со столбом (name)

Что вернула команда .schema?

Данная команда показала ка- кие 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:

О

```
.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
```

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> .timer on
sqlite> select count(*) from city;

count(*)
3

Run Time: real 0.001 user 0.000000 sys 0.000187
```

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

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.

```
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
```

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 | Елизово
sqlite> █
```

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;
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 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;
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 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.

The screenshot shows the Kaggle dataset page for "The Complete Pokemon Dataset" by Rounak Banik, updated 5 years ago. The dataset has 1025 downloads and a "New Notebook" button. The description states it contains data on more than 800 Pokemon from all 7 Generations. The page includes tabs for "Data Card", "Code (171)", and "Discussion (10)". The "About Dataset" section provides context, license (CC0: Public Domain), and expected update frequency (Not specified). A sidebar on the left contains navigation icons, and a bottom banner mentions cookie usage.

Рисунок 14 – Выбранный датасет с Kaggle

```
sqlite> select name, hp from pokemon where hp='50' order by name;
```

name	hp
Alolan Sandshrew	50
Aron	50
Azurill	50
Barboach	50
Bellsprout	50
Bonsly	50
Cacnea	50
Carbink	50
Cascoon	50
Chewtle	50
Clauncher	50
Cleffa	50
Clobbopus	50
Cloyster	50
Cubone	50
Deoxys Attack Forme	50
Deoxys Defense Forme	50
Deoxys Normal Forme	50
Deoxys Speed Forme	50
Diancie	50
Dottler	50
Dwebble	50
Fan Rotom	50
Frost Rotom	50
Galarian Meowth	50
Galarian Mr. Mime	50


```
sqlite> select name, total, legendary from pokemon where legendary="True" order by total;
```

name	total	legendary
Eternamax Eternatus	1125	True
Cosmog	200	True
Meltan	300	True
Kubfu	385	True
Cosmoem	400	True
Phione	480	True
Zygarde 10% Forme	486	True
Calyrex	500	True
Urshifu Single Strike Style	550	True
Gigantamax Urshifu Single Strike Style	550	True
Urshifu Rapid Strike Style	550	True
Gigantamax Urshifu Rapid Strike Style	550	True
Tapu Koko	570	True
Tapu Lele	570	True
Tapu Bulu	570	True
Tapu Fini	570	True
Articuno	580	True
Galarian Articuno	580	True
Zapdos	580	True
Galarian Zapdos	580	True
Moltres	580	True
Galarian Moltres	580	True
Raikou	580	True
Entei	580	True
Suicune	580	True
Regirock	580	True

```
sqlite> select name, type1, hp, attack, defense from pokemon where attack > 90 order by name;
```

name	type1	hp	attack	defense
Abomasnow	Grass	90	92	75
Alolan Graveler	Rock	55	95	115
Anorith	Rock	45	95	50
Basculin	Water	70	92	65
Cinccino	Normal	75	95	60
Cloyster	Water	50	95	180
Crustle	Bug	70	95	125
Cursola	Ghost	60	95	50
Deoxys Speed Forme	Psychic	50	95	90
Duraludon	Steel	70	95	115
Exeggutor	Grass	95	95	85
Exploud	Normal	104	91	63
Ferrothorn	Grass	74	94	131
Galarian Farfetch'd	Fighting	52	95	55
Garbodor	Poison	80	95	82
Gigantamax Duraludon	Steel	70	95	115
Gigantamax Garbodor	Poison	80	95	82
Gigantamax Toxtricity	Electric	75	98	70
Gliscor	Ground	75	95	125
Gourgeist Large Size	Ghost	75	95	122
Graveler	Rock	55	95	115
Greedent	Normal	120	95	95
Greninja	Water	72	95	67
Hawlucha	Fighting	78	92	75
Heatmor	Fire	85	97	66
Hitmontop	Fighting	50	95	95













Имя	Дата изменения	Тип	Размер
 1.csv	14.04.2023 12:08	Файл Microsoft E...	2 КБ
 1.json	14.04.2023 12:09	Исходный файл J...	3 КБ
 2.csv	14.04.2023 12:12	Файл Microsoft E...	2 КБ
 2.json	14.04.2023 12:13	Исходный файл J...	3 КБ
 3.csv	14.04.2023 12:17	Файл Microsoft E...	3 КБ
 3.json	14.04.2023 12:18	Исходный файл J...	7 КБ
 4.csv	14.04.2023 12:21	Файл Microsoft E...	29 КБ
 4.json	14.04.2023 12:22	Исходный файл J...	83 КБ
 5.csv	14.04.2023 12:23	Файл Microsoft E...	26 КБ
 5.json	14.04.2023 12:23	Исходный файл J...	67 КБ
 6.csv	14.04.2023 12:26	Файл Microsoft E...	26 КБ
 6.json	14.04.2023 12:26	Исходный файл J...	67 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`