

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
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

Кочкаров Умар Ахматович

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2023

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

**Ход работы:**

1. Создать общедоступный репозиторий с лицензией MIT и языком Python.

## Create a new repository

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

Owner \*



umarkochkarov ▾



Repository name \*

lb2.20 ✓

Great repository names are short and memorable. Need inspiration? How about [silver-pancake?](#)

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 ▾

This will set main as the default branch. Change the default name in your [settings](#).



You are creating a public repository in your personal account.

Create repository

Рисунок 1. Создание репозитория

2. Клонировать репозиторий на ПК:

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.20
$ git clone https://github.com/umarkochkarov/lb2.20.git
Cloning into 'lb2.20'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование репозитория

3. Организовать репозиторий в соответствии с моделью ветвления git-flow.

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/python/Ла62.20/lb2.20 (main)
$ 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/erken/Desktop/python/Ла62.20/lb2.20/.git/
hooks]
```

Рисунок 3. Организация репозитория в соответствии с моделью git-flow

4. Проработка примеров из лабораторной работы:

**Задание №1(7).** Выполнение команд.

```
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 4. Выполнение первого задания

**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 строка:

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

count (*)
1117

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

Рисунок 5. Выполнение второго задания

**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

Рисунок 6. Выполнение третьего задания

**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. Выполнение четвертого задания

**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.

```
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;
```

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

Рисунок 8. Выполнение пятого задания

**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.

```
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;
```

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

Рисунок 9. Выполнение шестого задания

**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 10. Выполнение седьмого задания 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 11. Выполнение шестого задания с «pipe» разделителем

5. **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.

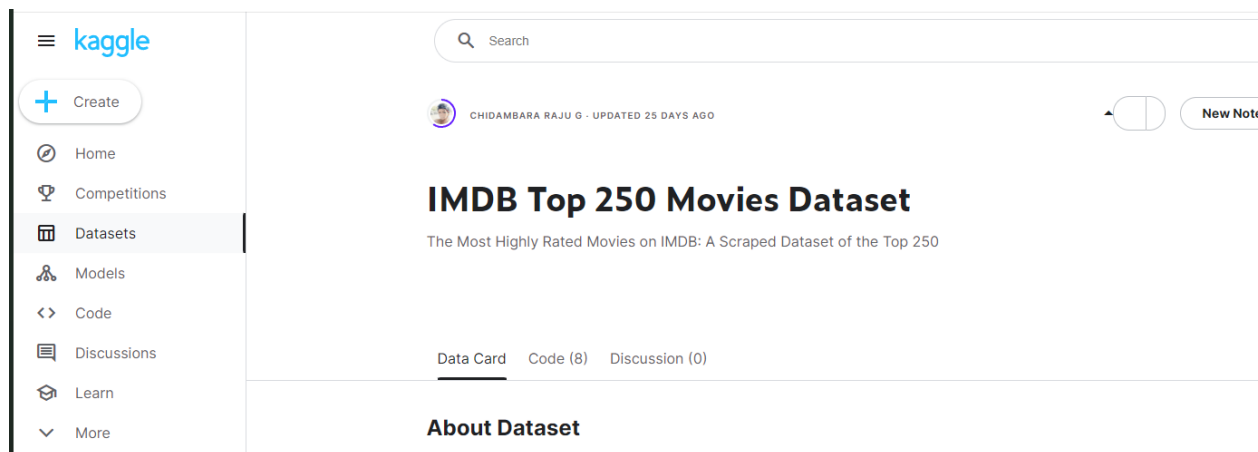


Рисунок 12. Выбранный датасет

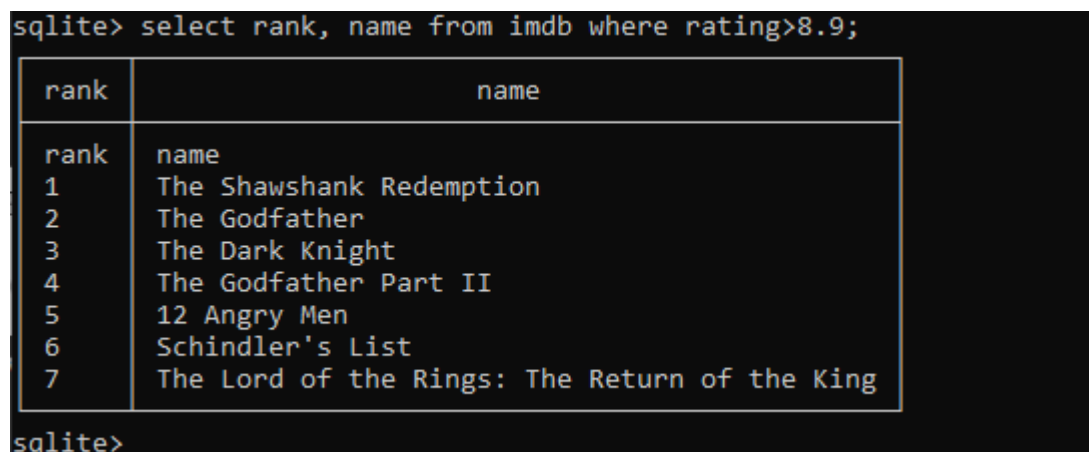


Рисунок 13. Первый запрос



Рисунок 14. Второй запрос

```
sqlite> select name, directors, tagline from imdb where year=2000 and rating>=8.0;
```

name	directors	tagline
Gladiator	Ridley Scott	Father of a murdered son, husband to a murdered wife and I s hall have my vengeance in this life or the next
Memento	Christopher Nolan	Some memories are best forgotten
Requiem for a Dream	Darren Aronofsky	From the director of [Pi]
Snatch	Guy Ritchie	Stealin' Stones and Breakin' Bones
Amores Perros	Alejandro G. Inarritu	Love. Betrayal. Death.

```
sqlite>
```

Рисунок 15. Третий запрос

```
sqlite> select name, year, rating from imdb where rating between 8.0 and 8.2 order by year;
```

name	year	rating
Sherlock Jr.	1924	8.2
The Gold Rush	1925	8.1
The General	1926	8.1
The Passion of Joan of Arc	1928	8.2
It Happened One Night	1934	8.1
Gone with the Wind	1939	8.2
Mr. Smith Goes to Washington	1939	8.1
The Wizard of Oz	1939	8.1
The Grapes of Wrath	1940	8.1
Rebecca	1940	8.1
To Be or Not to Be	1942	8.2
The Best Years of Our Lives	1946	8.1
The Treasure of the Sierra Madre	1948	8.2
The Third Man	1949	8.1
All About Eve	1950	8.2
Rashomon	1950	8.2
The Wages of Fear	1953	8.2
Tokyo Story	1953	8.2
Dial M for Murder	1954	8.2
On the Waterfront	1954	8.1
The Bridge on the River Kwai	1957	8.2
Wild Strawberries	1957	8.1
The Seventh Seal	1957	8.1
Some Like It Hot	1959	8.2
Ben-Hur	1959	8.1
The 400 Blows	1959	8.1
Yojimbo	1961	8.2
The Great Escape	1963	8.2
For a Few Dollars More	1965	8.2
The Sound of Music	1965	8.1
The Battle of Algiers	1966	8.1
Persona	1966	8.1
Cool Hand Luke	1967	8.1
The Exorcist	1973	8.1
Chinatown	1974	8.2
Monty Python and the Holy Grail	1975	8.2
Barry Lyndon	1975	8.1
Jaws	1975	8.1
Dersu Uzala	1975	8.2
Taxi Driver	1976	8.2
Rocky	1976	8.1
Network	1976	8.1
The Deer Hunter	1978	8.1

Рисунок 16. Четвертый запрос



```
sqlite> select name, year, rating, budget, rank from imdb where rating > 8.5 and budget < 50000000;
```

name	year	rating	budget	rank
The Shawshank Redemption	1994	9.3	25000000	1
The Godfather	1972	9.2	6000000	2
The Godfather Part II	1974	9.0	13000000	4
12 Angry Men	1957	9.0	350000	5
Schindler's List	1993	9.0	22000000	6
Pulp Fiction	1994	8.9	8000000	8
The Good, the Bad and the Ugly	1966	8.8	1200000	10
Star Wars: Episode V - The Empire Strikes Back	1980	8.7	18000000	15
Goodfellas	1990	8.7	25000000	17
One Flew Over the Cuckoo's Nest	1975	8.7	3000000	18
Se7en	1995	8.6	33000000	19
It's a Wonderful Life	1946	8.6	3180000	21
The Silence of the Lambs	1991	8.6	19000000	22
Life Is Beautiful	1997	8.6	20000000	26
Star Wars: Episode IV - A New Hope	1977	8.6	11000000	28
Spirited Away	2001	8.6	19000000	31

```
sqlite>
```

Рисунок 17. Пятый запрос

```
sqlite> select rank, name, rating from imdb where (directors='Christopher Nolan');
```

rank	name	rating
3	The Dark Knight	9.0
14	Inception	8.8
25	Interstellar	8.6
41	The Prestige	8.5
54	Memento	8.4
69	The Dark Knight Rises	8.4
127	Batman Begins	8.2

```
sqlite>
```

Рисунок 18. Шестой запрос

**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? SQL состоит из операторов, инструкций и вычисляемых функций.**

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: мини-мум, 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  
group 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`