

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

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

**Кафедра инфокоммуникаций**

**«Основы работы с SQLite3»**

**Отчет по лабораторной работе № 2.20(7)**

**по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Халимендик Я. Д. « » 2023г.

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. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования 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 \*** **Repository name \***

 Yana-Kh / Lab-2.20-OPJ

✓ Lab-2.20-OPJ is available.

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

**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:**

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: **Python**


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

**Choose a license**

License: **MIT License**

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

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

3. Выполните клонирование созданного репозитория.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git>git clone https://github.com/Yana-Kh/Lab-2.20-OPJ.git
Cloning into 'Lab-2.20-OPJ'...
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.
C:\Users\ynakh\OneDrive\Рабочий стол\Git>
```

Рисунок 2 – Клонирование репозитория

4. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ>git add .
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore

C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ>
```

Рисунок 3 – Дополнение файла .gitignore

5. Добавьте в файл README.md информацию о группе и ФИО студента, выполняющего лабораторную работу.

```
# Lab-2.20-OPJ
Выполнила: студентка ПИЖ-6-о-21-1
Халимендик Яна
```

Рисунок 4 – Дополнение файла README.md

6. Добавьте файл README и зафиксируйте сделанные изменения.

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ>git add .
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ>
```

Рисунок 5 – Фиксация изменений в репозитории

7. Решите задачу: выполните в песочнице команды:

```
create table customer(name);
```

```
select *
```

```
from customer;
```

```
.schema customer
```

Что вернула команда .schema?

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ\ex>sqlite3 ex1.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read 1.sql
CREATE TABLE customer(name);
sqlite> .quit
```

Данная команда вернула 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. Если ее включить, в результатах запроса добавится строчка:

Run Time: real XXX user XXX sys XXX

Например:

```
sqlite> .SOMETHING on
```

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

Какая команда должна быть вместо SOMETHING?

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ\ex>sqlite3 ex2.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read 2.sql
Run Time: real 0.001 user 0.000000 sys 0.000000
Parse error near line 2: no such table: city
sqlite> .mode box
sqlite> .import --csv city.csv city
sqlite> .read 2.sql

count(*)
1117

Run Time: real 0.002 user 0.000000 sys 0.000000
sqlite>
```

Ответ: timer

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

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

Затем выполните такой запрос:

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

Какое число он вернул?

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ\ex>sqlite3 ex3.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read 3.sql

max(length(city))
25

sqlite> _
```

Ответ: 25

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.

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

```
do_something
```

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

Какая команда должна быть вместо `do_something` ?

Ответ: `.mode csv`

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:

timezone	city_count

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

Укажите в ответе значение city\_count для timezone = UTC+5.

```
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ\ex>sqlite3 ex5.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read 5.sql
```

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

```
sqlite> .quit
C:\Users\ynakh\OneDrive\Рабочий стол\Git\Lab-2.20-OPJ\ex>_
```

Ответ: 58

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.

Например:

Нижний Новгород, Москва, Владивосток

Чтобы посчитать расстояние между двумя городами, используйте формулу из школьного курса геометрии:

$$distance^2 = (lat_1 - lat_2)^2 + (lon_1 - lon_2)^2$$

Где  $(lat_1, lon_1)$  – координаты первого города, а  $(lat_2, lon_2)$  – координаты второго.

```
sqlite> .read 6.sql
```

city	distance
Новокуйбышевск	0.18569700863441
Чапаевск	0.358068603404667
Кинель	0.528066220190501

```
sqlite>
```

Ответ: Новокуйбышевск, Чапаевск, Кинель

13. Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию. Получится примерно так:

timezone	city_count
UTC+3	xxx
UTC+5	xxx
UTC+7	xxx
UTC+4	xxx
...	

```
C:\> Командная строка - sqlite3 test.db
Enter ".help" for usage hints.
sqlite> .read 7.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> .quit
```

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

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

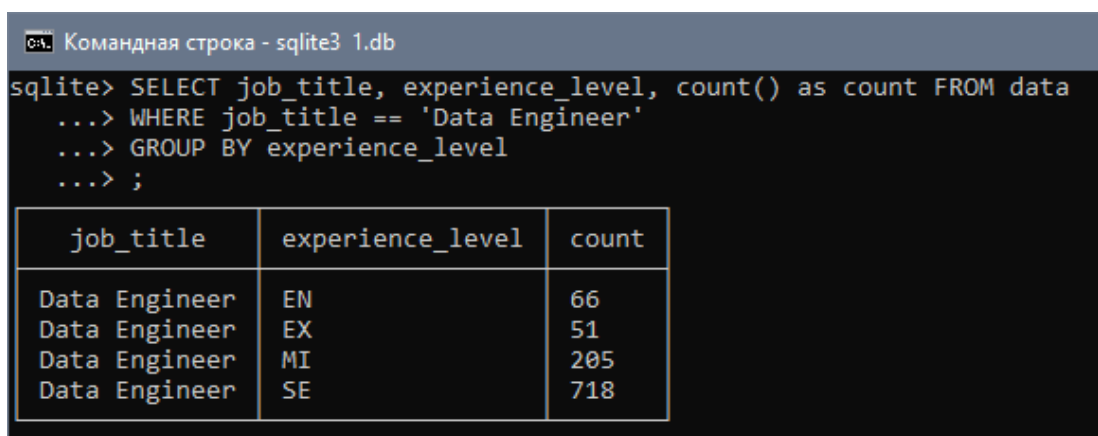
Как выглядит четвертая строка результата?

```
sqlite> .read 7.sql
timezone|count()
UTC+3|1320
UTC+5|346
UTC+7|172
UTC+4|132
UTC+9|62
UTC+8|56
UTC+2|44
UTC+10|44
UTC+11|34
UTC+6|12
UTC+12|12
timezone|1
sqlite>
```

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

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.



The screenshot shows a SQLite command prompt window titled "Командная строка - sqlite3 1.db". The user has entered the following SQL query:

```
sqlite> SELECT job_title, experience_level, count() as count FROM data
...> WHERE job_title == 'Data Engineer'
...> GROUP BY experience_level
...> ;
```

The result is displayed as a table with three columns: job\_title, experience\_level, and count. The data is as follows:

job_title	experience_level	count
Data Engineer	EN	66
Data Engineer	EX	51
Data Engineer	MI	205
Data Engineer	SE	718

Рисунок 1 – Запрос «Количество инженеров данных по уровню опыта»



```
sqlite> SELECT job_title, company_size, remote_ratio
...> FROM data
...> WHERE remote_ratio >= 50 AND company_size == 'L'
...> LIMIT 10;
```

job_title	company_size	remote_ratio
Machine Learning Engineer	L	50
AI Developer	L	50
Research Scientist	L	50
Business Data Analyst	L	50
Machine Learning Engineer	L	50
Data Scientist	L	50
Lead Data Analyst	L	50
Data Scientist	L	50
AI Scientist	L	50
Data Analyst	L	50

```
sqlite>
```

Рисунок 2 – Запрос «Должности в больших компаниях, где объем работы выполняемый удаленно выше 50%»

```
sqlite> read 3.sql
```

experience_level	min_salary	max_salary	potention
EN	10000	96000	86000
EX	100000	85000	-15000
MI	10000	99100	89100
SE	10000	99750	89750

```
sqlite> read 3.sql
```

Рисунок 3 – Запрос «Минимальная, максимальная зарплата и их разница по уровню опыта»

year	average_salary
2023	149045.541
2022	133338.621
2021	94087.209
2020	92302.632

Рисунок 4 – Запрос «Средняя зарплата по году (округление до тысячных)»

job_title	sum of type
Data DevOps Engineer	50000
Data Management Specialist	65000
Data Strategist	162000
Data Analytics Consultant	163000
Data Analytics Specialist	190000
Data Modeler	237800
Data Scientist Lead	268000
Data Operations Analyst	362250
Data Science Tech Lead	375000
Data Analytics Engineer	395000

Рисунок 5 – Запрос «Суммарная заработная плата по должностям, начинающимся с ‘Data’»

10. Зафиксируйте сделанные изменения в репозитории.

 Yana-Kh id	
 ex	ex
 id	id
 .gitignore	ignore + read_me
 LICENSE	Initial commit
 README.md	ignore + read_me

Рисунок 10 – Фиксирование изменений в репозитории

Вопросы для защиты работы:

1. Каково назначение реляционных баз данных и СУБД?

Реляционные базы данных используются для хранения, организации, управления и доступа к данным в приложениях и информационных системах. Они организуют данные в таблицы, состоящие из строк и столбцов, где каждая строка представляет собой отдельную запись, а каждый столбец представляет собой отдельный атрибут.

## 2. Каково назначение языка SQL?

Язык SQL (Structured Query Language) является стандартным языком запросов для работы с данными в реляционных базах данных и предоставляет мощные возможности для создания, изменения, извлечения и управления данными.

## 3. Из чего состоит язык SQL?

Язык SQL состоит из операторов, инструкций и вычисляемых функций. Зарезервированные слова, которыми обычно выступают операторы, принято писать заглавными буквами.

## 4. В чем отличие СУБД SQLite от клиент-серверных СУБД?

Отличие между СУБД SQLite и клиент-серверными СУБД:

- SQLite: Встраиваемая СУБД, работает локально внутри приложения, не требует отдельного сервера. Ограничен однопользовательским доступом и не масштабируется для больших проектов.
- Клиент-серверные СУБД: Разделение на клиентскую и серверную части. Сервер управляет базой данных, клиенты подключаются удаленно по сети. Поддерживает многопользовательский доступ и масштабируется для обработки больших объемов данных.

## 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 (`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?

В реляционных базах данных, первичный ключ (Primary Key) – это уникальное идентифицирующее поле или набор полей в таблице. Он служит для однозначной идентификации каждой записи (строки) в таблице.

10. Как сделать первичный ключ таблицы автоинкрементным?

Для создания автоинкрементного первичного ключа в таблице SQLite, вы можете использовать тип данных INTEGER и атрибут AUTOINCREMENT.

Пример:

```
CREATE TABLE your_table_name (  
    id INTEGER PRIMARY KEY AUTOINCREMENT...);
```

11. Каково назначение инструкций NOT NULL и DEFAULT при создании таблиц?

Инструкция NOT NULL указывает, что столбец не может содержать NULL (пустое) значение, обеспечивая целостность данных.

Инструкция DEFAULT позволяет установить значение по умолчанию для столбца, которое будет использоваться, если явно не указано другое значение при вставке данных.

12. Каково назначение внешних ключей в таблице? Как создать внешний ключ в таблице?

Внешние ключи в таблице используются для установления связей между двумя таблицами в реляционных базах данных. Они определяют отношения между записями в разных таблицах.

Пример:

```
CREATE TABLE pages (  
    ...  
    FOREIGN KEY (theme) REFERENCES sections(_id)  
    ...);
```

13. Как выполнить вставку строки в таблицу базы данных SQLite?

С помощью оператора INSERT.

```
INSERT INTO <table_name>  
(<column_name1>, <column_name2>, ...)
```

## VALUES

(<value1>, <value2>, ...);

### 14. Как выбрать данные из таблицы SQLite?

С помощью оператора SELECT.

SELECT \* FROM ;

### 15. Как ограничить выборку данных с помощью условия WHERE?

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?

С помощью оператора ORDER BY.

ORDER BY column1 ASC/DESC, column2 ASC/DESC, ...

### 17. Как выполнить обновление записей в таблице SQLite?

Для обновления записей в таблице SQLite используйте оператор UPDATE с указанием имени таблицы, столбцов и новых значений, а также условия WHERE для определения, какие строки обновить.

UPDATE your\_table

SET column1 = new\_value1, column2 = new\_value2;

### 18. Как удалить записи из таблицы SQLite?

Для удаления записей из таблицы SQLite вы можете использовать оператор DELETE. Он позволяет удалить одну или несколько строк, удовлетворяющих заданному условию.

DELETE FROM your\_table;

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?

Подзапросы в SQLite позволяют выполнять вложенные запросы внутри основного запроса для выполнения дополнительных вычислений, фильтрации или связывания данных.

Шаблоны в SQLite позволяют определить временную таблицу, которая может быть использована внутри запроса для создания и манипулирования данными во время выполнения запроса.

Подзапросы используются как внутренние выражения внутри операторов SELECT, FROM, WHERE, HAVING и других частей запроса.

Шаблоны определяются с помощью оператора WITH и могут быть использованы в основном запросе, как обычная таблица.

Оба инструмента предоставляют дополнительные возможности для обработки данных в SQLite, позволяя выполнять более сложные операции и повышая гибкость запросов.

### 23. Каково назначение представлений VIEW в SQLite?

Представления (VIEW) в SQLite создают виртуальные таблицы, которые представляют результат выполнения запроса. Они упрощают выполнение сложных запросов, абстрагируют данные, обеспечивают безопасность, позволяют повторное использование запросов и обновление данных.

### 24. Какие существуют средства для импорта данных в SQLite?

```
.import --csv data.csv data
```

### 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?

С помощью ORDER BY и GROUP BY.

### 27. Каково назначение "табличных выражений" в SQLite?

Табличные выражения в SQLite представляют результаты выполнения подзапросов или вложенных запросов, которые можно использовать как виртуальные таблицы в основном запросе.

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



Устанавливается вид данных с помощью `.mode csv/json/...`, направляет вывод результатов запроса в файл `.once data.csv` для одного запроса и `.output data.csv`, действующий, пока не произведется отмена.

29. Какие еще форматы для экспорта данных Вам известны?

SQL-скрипты, JSON, XML