

**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

Рязанцев.М.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. Решите задачу: выполните в песочнице команды:

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

```
select *
```

```
from customer;
```

```
.schema customer
```

Что вернула команда .schema?

```
D:\2kurs\!22kurs\opi_norm\opi_lab220\exmpl>sqlite3 ex1.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read ex1.sql
Parse error near line 1: table customer already exists
CREATE TABLE customer(name);
      ^--- error here
CREATE TABLE customer(name);
sqlite>
```

Данная команда вернула список и структуру всех таблиц в базе.

2. 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:

```
Run Time: real XXX user XXX sys XXX
```

Например:

```
sqlite> .SOMETHING on
```

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

Какая команда должна быть вместо SOMETHING?

```
D:\2kurs\!22kurs\opi_norm\opi_lab220\exmpl>sqlite3 ex2.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read ex2.sql
Run Time: real 0.000 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 ex2.sql



|          |
|----------|
| count(*) |
| 1117     |



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

Ответ: timer

3. 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?

```
D:\2kurs\!22kurs\opi_norm\opi_lab220\exmpl>sqlite3 ex3.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read ex3.sql

max(length(city))
25

sqlite>
```

Ответ: 25

4. 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`

5. 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.

```
D:\2kurs\!22kurs\opi_norm\opi_lab220\exmpl>sqlite3 ex5.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read ex5.sql
```

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

Ответ: 58

6. 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.

```
D:\2kurs\!22kurs\opi_norm\opi_lab220\exmpl>sqlite3 ex6.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .read ex6.sql
```

city	distance
??????????????	0.18569700863441
?????????	0.358068603404667
??????	0.528066220190501

Ответ: Новокуйбышевск, Чапаевск, Кинель

7. 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
...	

```
D:\2kurs\!22kurs\opi_norm\opi_la
SQLite version 3.41.2 2023-03-22
Enter ".help" for usage hints.
sqlite> .read ex7.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

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

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

Как выглядит четвертая строка результата?

```
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
```

8. 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.

```
sqlite> .read id1.sql
```

publisher	society	count
Kluwer	no	16

Рисунок 1 – Запрос «Количество не общественных журналов Kluwer»

```
sqlite> .read id2.sql
```

title	society	citations
Economic Record	yes	188
Review of Income and Wealth	yes	189
Journal of Risk and Insurance	yes	193
Journal of Agricultural Economics	yes	220
Journal of Economic Issues	yes	355
European Economic Review	yes	1243
Journal of Economic Literature	yes	1530
Journal of Economic Perspectives	yes	1583
Journal of the American Statistical Association	yes	2800
Journal of Finance	yes	3791

Рисунок 2 – Запрос «Общественные журналы с кол-вом цитат меньше 4000»

```
sqlite> .read id3.sql
```

mins	maxs	razs	publisher
518	518	0	ANU Press
1058	803	-255	Academic Press
277	277	0	Agric. Econ. Society
1867	940	-927	Am Ec Assn
510	510	0	Am Stat Assn
1253	1253	0	Am. Ag. Econ Assn
2272	2272	0	Am. Finance Assn
720	720	0	Am. Risk & Ins. Assn
1260	1260	0	Am. Statistical Assn
1003	1003	0	Assn Ev. Economics
1247	969	-278	Blackwell
394	394	0	Brookings Inst.
1200	899	-301	Cambridge Univ Pres
1711	911	-800	Carfax
448	448	0	Dekker
767	767	0	Duke Univ Press
421	421	0	Ec. Society of Australia
1110	940	-170	Elsevier
500	500	0	Frank Cass
725	725	0	George Mason Univ

Рисунок 3 – Запрос «Минимальное, максимальное кол-во страниц их разница у автора»


```
sqlite> .read id4.sql
```

publisher	sred_pages
Assn Ev. Economics	1003.0
Southern Econ. Assn	1032.0
Georgia State Univ	1067.0
Inst for OR and MS	1175.0
Oxford Univ. Press	1001.0
Am. Ag. Econ Assn	1253.0
Am. Statistical Assn	1260.0
Am. Finance Assn	2272.0
Am Ec Assn	1813.0
Agric. Econ. Society	277.0
Wiley	644.6
So Afr ec history assn	309.0
Academic Press	815.8
Springer	649.9
Sage	510.5
Brookings Inst.	394.0
LSE and Univ of Bath	406.0
Ec. Society of Australia	421.0
Heldref	426.0
Blackwell	756.3
Intl Atlantic Ec. Soc.	447.0
Dekker	448.0
Frank Cass	500.0
Univ Wash Press	500.0
Am Stat Assn	510.0
Carfax	1045.7
ANU Press	518.0
Inst of Devel Econ	539.0
M.E Sharpe	550.0
Oxford Univ Press	647.4
Kluwer	789.7
Pop Assn America	568.0
Routledge	918.8
Univ of Wisconsin Press	708.5
Int Assn for Res in I & W	585.0
Cambridge Univ Pres	768.5
MIT Press	602.0
Helbing & Lichtenhahn	618.0
University of Toronto Press	714.0
Am. Risk & Ins. Assn	720.0
George Mason Univ	725.0

Рисунок 4 – Запрос «Средняя кол-во страниц у автора»

```
sqlite> .read id5.sql
```

publisher	mfy	razs
Mohr Siebeck	1844	34
Kluwer	1852	-269
MIT press	1886	-734
Blackwell	1890	-278
Univ of Chicago Press	1892	-780
Am Ec Assn	1911	-927
Am. Ag. Econ Assn	1918	0
Springer	1922	-774
Ec. Society of Australia	1925	0
Univ of Wisconsin Press	1925	257
Southern Econ. Assn	1932	0
Cambridge Univ Pres	1941	-301
Routledge	1943	-909
Am. Finance Assn	1945	0
Int Assn for Res in I & W	1945	0
MIT	1947	65
Agric. Econ. Society	1948	0
Elsevier	1948	-170
Helbing & Lichtenhahn	1948	0
National Tax Assn.	1948	0
Oxford Univ. Press	1949	-468
Inst for OR and MS	1954	0
Carfax	1960	-800
Oxford Univ Press	1962	248
Academic Press	1963	-255
Inst of Devel Econ	1963	0
ANU Press	1964	0
Am. Risk & Ins. Assn	1964	0
Frank Cass	1964	0
Pop Assn America	1964	0
Univ Wash Press	1966	0
Assn Ev. Economics	1967	0
LSE and Univ of Bath	1967	0
Duke Univ Press	1969	0
Ohio State Univ. Press	1969	0
Brookings Inst.	1970	0
Georgia State Univ	1970	0
Heldref	1970	0
RAND	1970	0
Am. Statistical Assn	1971	0

Рисунок 5 – Запрос «Год издания первой работы автора и разница по кол-ву страниц его работ»

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