

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Лабораторная работа 2.20**

**Основы работы с SQLite3**

Выполнил студент группы ИВТ-б-о-20-1

Павленко М.С. «    » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

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

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

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

Ход работы

Задание 1. Выполнение команды.

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

Рисунок 1. Создание таблицы customer со столбцом name

Задание 2. Решите задачу: с помощью команды. help найдите в песочнице команду, которая отвечает за вывод времени выполнения запроса.

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


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

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


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

  
Run Time: real 0.001 user 0.000000 sys 0.000000  
sqlite> |
```

Рисунок 2. Время выполнения запроса

Задание 3. Решите задачу: загрузите файл city.csv. Затем выполните такой запрос:

```
sqlite> select max(length(city)) from city
...> ;
```

max(length(city))
25

```
Run Time: real 0.001 user 0.000000 sys 0.000000
sqlite> |
```

Рисунок 3. Вывод запроса

Задание 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:

```
sqlite> .mode csv
sqlite> .import city.csv city
```

Рисунок 4. Добавления данных без использования опции csv

Задание 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:

```
sqlite> select timezone,
...> count(*) city_count
...> from city
...> where federal_district
...> in ('Сибирский', 'Приволжский')
...> group by 1
...> order by 1 asc;
```

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

Рисунок 5. Результат запроса

Задание 6. Решите задачу: напишите в песочнице запрос, который найдет три ближайших к Самаре города, не считая саму Самару.

```
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 6. Запрос

```
0.001052999999999886 | Заречный  
0.00948430000000004 | Каменка  
0.01199310000000051 | Елизovo
```

Рисунок 7. Результат запроса

Задание 7. Решите задачу: напишите в песочнице запрос, который посчитает количество городов в каждом часовом поясе. Отсортируйте по количеству городов по убыванию.

```
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

```
sqlite>
```

Рисунок 8. Результат запроса

Индивидуальное задание:

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

count(*)
312

Рисунок 9. Первый запрос

```
sqlite> select Max(Score) from happy;
```

Max(Score)
7.769

```
sqlite> select "Country or region", Year FROM happy WHERE Score=7.769;
```

Country or region	Year
Finland	2019

```
sqlite> |
```

Рисунок 10. Второй запрос (Страна и год с самым высоким индексом счастья)

```
sqlite> select "Country or region", Year FROM happy WHERE Score=6.774;
```

Country or region	Year
South Sudan	2019

```
sqlite> |
```

Рисунок 11. (Страна и год с самым низким индексом счастья)

```
sqlite> select "Country or region", Year, Score FROM happy;
```

Country or region	Year	Score
United Arab Emirates	2018	6.774

Рисунок 12. Четвёртый запрос (Страна с самым высоким ВВП на душу населения и её уровень счастья)

```
sqlite> select "Country or region", year, Score from happy where year in ('2018') order by 3 DESC limit 10;
```

Country or region	Year	Score
Finland	2018	7.632
Norway	2018	7.594
Denmark	2018	7.555
Iceland	2018	7.495
Switzerland	2018	7.487
Netherlands	2018	7.441
Canada	2018	7.328
New Zealand	2018	7.324
Sweden	2018	7.314
Australia	2018	7.272

Рисунок 13. Пятый запрос (Топ 10 стран с самым высоким уровнем счастья в 2018 году)

Вывод: в ходе работы были исследованы базовые возможности системы управления базами данных SQLite3.

#### Контрольные вопросы

1. Каково назначение реляционных баз данных и СУБД? Реляционные базы данных применяются для отслеживания товарных запасов, обработки торговых транзакций через Интернет, управления большими объемами критически важных данных заказчиков

2. Каково назначение языка SQL? это язык программирования, предназначенный для работы с наборами фактов и отношениями между ними. В программах управления реляционными базами данных, таких как Microsoft Office Access, язык SQL используется для работы с данными.

3. Из чего состоит язык SQL? Язык SQL представляет собой совокупность операторов, инструкций, вычисляемых функций. Согласно общепринятому стилю программирования, операторы (и другие зарезервированные слова) в SQL обычно рекомендуется писать прописными буквами. Операторы SQL делятся на: операторы определения данных (Data Definition Language, DDL)

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

5. Как установить SQLite в Windows и Linux? SQLite славится своей отличной нулевой конфигурацией, что означает, что не требуется сложной настройки или администрирования. В этой главе мы рассмотрим процесс настройки SQLite в Windows, Linux и Mac OS X.

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`. Если вы работаете с одной БД, а потом открываете другую, то текущей становится вторая БД.

8. Как создать и удалить таблицу в SQLite? Для создания таблиц используется команда `CREATE TABLE`. Для удаления таблицы применяется команда `DROP TABLE`.

9. Что является первичным ключом в таблице? В реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа (или ключа по умолчанию).

10. Как сделать первичный ключ таблицы автоинкрементным? Создание первичного ключа автоматически приводит к созданию соответствующего уникального кластеризованного индекса (или некластеризованного при наличии такого указания).

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

12. Каково назначение внешних ключей в таблице? Как создать внешний ключ в таблице? Внешние ключи позволяют установить связи между таблицами. Внешний ключ устанавливается для столбцов из зависимой, подчиненной таблицы, и указывает на один из столбцов из главной таблицы. Как правило, внешний ключ указывает на первичный ключ из связанной главной таблицы. Для создания ограничения внешнего ключа после `FOREIGN KEY` указывается столбец таблицы, который будет представлять внешний ключ. А после ключевого слова `REFERENCES` указывается имя связанной таблицы, а затем в скобках имя связанного столбца, на который будет указывать внешний ключ. После выражения `REFERENCES` идут выражения `ON DELETE` и `ON UPDATE`, которые задают действие при удалении и обновлении строки из главной таблицы соответственно.

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

записи мы поговорим о добавлении данных и строк в таблицы реляционных баз данных. Естественно, рассматривать то, как добавить строку в таблицу базы данных мы будем на примере SQLite3, как и любые примеры в данном разделе. Напомню, что для добавления данных в таблицу в SQL есть команда INSERT, ранее мы очень подробно рассмотрели синтаксис команды INSERT в SQLite, теперь давайте попрактикуемся и научимся добавлять строки в таблицы баз данных под управлением SQLite3.

14. Как выбрать данные из таблицы SQLite? Получать все строки с помощью с помощью `cursor.fetchall()`. Использовать `cursor.fetchmany(size)` для получения ограниченного количества строк, а также `cursor.fetchone()` — для одной. Использовать переменные Python в запросе SQLite для передачи динамических данных.

15. Как ограничить выборку данных с помощью условия WHERE? Кроме фильтров и [сотрировок](#), в SELECT запросах можно также ограничивать выборку. Например, когда вам нужны не все данные, а скажем только первые 10 строк. Для этого существует конструкция LIMIT и давайте попробуем вывести 5 самых дорогих товаров.

16. Как упорядочить выбранные данные? Выделите ячейку в столбце, который требуется отсортировать. На вкладке Данные в группе Сортировка и фильтр нажмите кнопку Сортировка. В диалоговом окне Сортировка в группе Столбец в поле Сортировать по выберите столбец, по которому необходимо выполнить сортировку.

17. Как выполнить обновление записей в таблице SQLite? Сперва нужно установить SQLite-соединение из Python.

Дальше необходимо создать объект `cursor` с помощью объекта соединения.

После этого – создать запрос UPDATE. ...

Дальше запрос выполняется с помощью `cursor`.

18. Как удалить записи из таблицы SQLite? DELETE FROM COMPANY WHERE ID;

19. Как сгруппировать данные из выборке из таблицы SQLite?

20. Как получить значение агрегатной функции (например: минимум, максимум, количество



записей и т. д.) в выборке из таблицы SQLite?

21. Как выполнить объединение нескольких таблиц в операторе SELECT?

Группировка данных при выборке данных реализуется при помощи команды SELECT, уточняющей фразы GROUP BY и функций агрегации. Ключевое слово GROUP BY необязательное, но, если вы его используете, оно должно следовать за уточняющей фразой WHERE.

22. Каково назначение подзапросов и шаблонов при работе с таблицами SQLite?

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?

25. Каково назначение команды .schema ? Показывает схему таблицы

26. Как выполняется группировка и сортировка данных в запросах SQLite? группировку данных выборки, которая реализуется при помощи команды SELECT и уточняющей фразы GROUP BY.

27. Каково назначение "табличных выражений" в SQLite? Для обобщённых табличных выражений (Common Table Expression, CTE), позволяющих использовать временные именованные результирующие наборы, задаваемые при помощи оператора WITH, разрешён выбор режимов "MATERIALIZED" и "NOT MATERIALIZED".

28. Как осуществляется экспорт данных из SQLite в форматы CSV и JSON? Консоль — это киллер-фича SQLite: более мощный инструмент анализа данных, чем Excel, и сильно более простой, чем какой-нибудь pandas. Данные из CSV загружаются одной командой, таблица создается автоматически: Неважно, насколько развесистый JSON — можно выбрать атрибуты любой вложенности: