

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №2.20

Дисциплина: «Анализ данных»
Тема: «Основы работы с SQLite3»

Выполнил:
Епифанов Алексей Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных систем
», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Цель: исследовать базовые возможности системы управления базами данных SQLite3.

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.

2. Решил задачу 1: выполните в песочнице команды:

```
create table customer(name);
select *
from customer;
.schema customer
```

Что вернула команда .schema?

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

Рисунок 1. Выполнение задачи

Ответ: .schema вернула CREATE TABLE customer(name);.

3. Решил задачу 2: 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.

```
.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
.width NUM1 NUM2 ...  Set minimum column widths for columnar output
sqlite> .timer on
sqlite> select count(*) from city;
1117
Run Time: real 0.000 user 0.000126 sys 0.000085
sqlite> █
```

Рисунок 2. Выполнение задачи

Ответ: нужная команда - .timer on.

4. Решил задачу 3:

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?

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

Рисунок 3. Выполнение задачи

Ответ: запрос вернул число 25.

5. Решил задачу 4: 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:

```
.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> select max(length(city)) from city;
25
sqlite> █
```

Рисунок 4. Выполнение задачи

Ответ: необходимо перед загрузкой написать команду `.mode csv`.

6. Решил задачу 5: 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. Укажите в ответе значение `city_count` для `timezone = UTC+5`.

```
sqlite> .open city-1.db
sqlite> .mode box
sqlite> .header on
sqlite> SELECT timezone, COUNT(*) AS city_count
...> FROM city
...> WHERE federal_district IN ('Сибирский', 'Приволжский')
...> GROUP BY timezone
...> ORDER BY timezone;
```

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

Рисунок 5. Выполнение задачи

Ответ: в часовом поясе UTC+5 58 городов Сибирского и Приволжского федеральных округов.

7. Решил задачу 6: 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 SamaraCoords AS (
...>     SELECT geo_lat, geo_lon
...>     FROM city
...>     WHERE city = 'Самара'
...> )
...> SELECT GROUP_CONCAT(city, ', ') AS nearest_cities
...> FROM (
...>     SELECT city
...>     FROM city, SamaraCoords
...>     WHERE city.city <> 'Самара'
...>     ORDER BY (POWER(city.geo_lat - SamaraCoords.geo_lat, 2) +
...>                POWER(city.geo_lon - SamaraCoords.geo_lon, 2))
...>     LIMIT 3
...> );
Новокуйбышевск, Чапаевск, Кинель
sqlite>

```

Рисунок 6. Выполнение задачи

Ответ: 3 ближайших города в порядке удаления от Самары: Новокуйбышевск, Чапаевск, Кинель.

8. Решил задачу 7: 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 заголовками, с разделителем «pipe» |
Как выглядит четвертая строка результата?

```

Last login: Sun Mar 31 19:35:05 2024 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> .open city-1.db
sqlite> .mode box
sqlite> .header on
sqlite> SELECT timezone, COUNT(*) as city_count
...> FROM city
...> GROUP BY timezone
...> ORDER BY city_count 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> .mode csv
sqlite> .separator |
sqlite> SELECT timezone, COUNT(*) as city_count
...> FROM city
...> GROUP BY timezone
...> ORDER BY city_count 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 7. Выполнение задачи

Ответ: 4 строка результата выглядит - UTC+4|66

9. Выполнил 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.

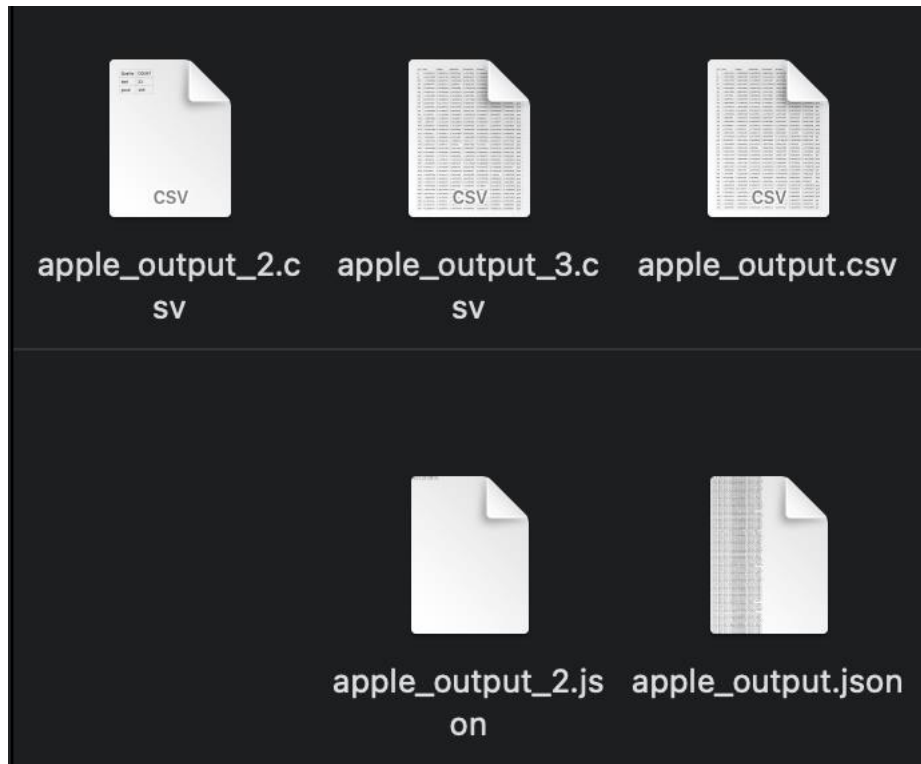
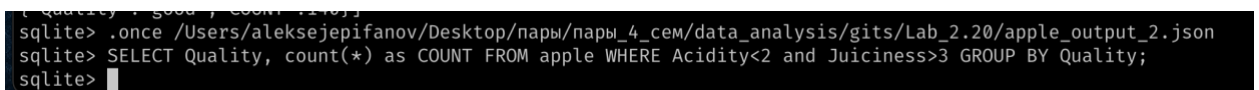


Рисунок 8. Выполнение задачи



Риснок 9. Для выгрузки используется .once

Контрольные вопросы:

1. Каково назначение реляционных баз данных и СУБД?

Слово "реляция" происходит от "relation", то есть "отношение". Это означает, что в РБД существуют механизмы установления связей между таблицами. Делается это с помощью так называемых первичных и внешних ключей.

2. Каково назначение языка SQL?

SQL – это язык программирования декларативного типа. В отличие от привычных нам процедурных языков, в которых есть условия, циклы и функции, в декларативных языках подобных алгоритмических конструкций

почти нет. Декларативные выражения представляют собой скорее запросы, описание того, что хочет получить человек.

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. При этом ключевыми операциями являются создание таблиц, добавление записей в таблицы, изменение и удаление записей, выборка записей из таблиц, изменение структуры таблиц.

3. Из чего состоит язык SQL?

Сам язык SQL состоит из операторов, инструкций и вычисляемых функций. Зарезервированные слова, которыми обычно выступают операторы, принято писать заглавными буквами. Однако написание их не прописными, а строчными буквами к ошибке не приводит.

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

SQLite – это система управления базами данных, отличительной особенностью которой является ее встраиваемость в приложения. Это значит, что большинство СУБД являются самостоятельными приложениями, взаимодействие с которыми организовано по принципу клиент-сервер.

Программа-клиент посылает запрос на языке SQL, СУБД, которая в том числе может находиться на удаленном компьютере, возвращает результат запроса.

В свою очередь SQLite является написанной на языке C библиотекой, которую динамически или статически подключают к программе. Для большинства языков программирования есть свои привязки (API) для библиотеки SQLite. Так в Python СУБД SQLite импортируют командой `import sqlite3`. Причем модуль `sqlite3` входит в стандартную библиотеку языка и не требует отдельной установки.

5. Как установить SQLite в Windows и Linux?

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).

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?

Первичный ключ в таблице базы данных — это уникальный идентификатор для каждой строки таблицы. Он гарантирует уникальность записи и используется для создания связей между таблицами (внешних ключей).

В архитектуре реляционной базы данных, первичный ключ может состоять из одного столбца (простой или одиночный ключ) или может комбинировать несколько столбцов (составной ключ), если уникальность

может быть гарантирована только их комбинацией. Значения первичного ключа должны быть уникальными и не могут быть NULL.

10. Как сделать первичный ключ таблицы автоинкрементным?

Если нам не важно, какие конкретно идентификаторы будут записываться в поле `_id`, а важна только уникальность поля, следует назначить полю еще один ограничитель – автоинкремент – `AUTOINCREMENT`.

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

Ограничитель `NOT NULL` используют, чтобы запретить оставление поля пустым. По умолчанию, если поле не является первичным ключом, в него можно не помещать данные. В этом случае полю будет присвоено значение `NULL`. В случае `NOT NULL` вы не сможете добавить запись, не указав значения соответствующего поля.

Однако, добавив ограничитель `DEFAULT`, вы сможете не указывать значение. `DEFAULT` задает значение по умолчанию. В результате, когда данные в поле не передаются при добавлении записи, поле заполняется тем, что было указано 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. Внешний ключ в одной таблице для другой является первичным. Внешние ключи не обязаны быть уникальными. В одной таблице, может быть, несколько внешних ключей, при этом каждый будет устанавливать связь со своей таблицей, где он является первичным.

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

С помощью оператора `INSERT` языка `SQL` выполняется вставка данных в таблицу.

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

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

14. Как выбрать данные из таблицы SQLite?

С помощью оператора SELECT.

15. Как ограничить выборку данных с помощью условия WHERE?

WHERE позволяет задавать условие, согласно которому отображаются только удовлетворяющие ему строки.

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. ASC – сортировка от меньшего значения к большему. DESC – сортировка от большего значения к меньшему.

17. Как выполнить обновление записей в таблице SQLite?

UPDATE ... SET – обновление полей записи

UPDATE имя_таблицы

SET имя_столбца = новое_значение

WHERE условие;

18. Как удалить записи из таблицы SQLite?

DELETE FROM – удаление записей таблицы

DELETE FROM имя_таблицы WHERE условие;

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?

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?

Группировка – GROUP BY, сортировка – ORDER BY.

27. Каково назначение "табличных выражений" в SQLite?

Табличные выражения в SQLite используются для структурирования запросов и улучшения их читаемости и гибкости. Табличные выражения можно считать временными наборами записей, которые доступны в пределах одного запроса.

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

`.mode csv` или `.mode json`

`.once <имя файла> или .output`

любой запрос

В результате вывод запроса уйдет в выбранный файл.

29. Какие еще форматы для экспорта данных Вам известны?

`.mode list, .mode markdown, .mode html ...`

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