ЛАБОРАТОРНАЯ РАБОТА 5.1

ВВЕДЕНИЕ В СУБД MONGODB. УСТАНОВКА MONGODB. НАЧАЛО РАБОТЫ С БД

Выполнил: Рыбалко Олег К32392

Цель: овладеть практическими навыками установки СУБД MongoDB.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД MongoDB 4+, 6.0.6 (текущая)

Практическое задание:

1. Установка

Для установки mongodb мы воспользуемся Docker и конфигурацией docker compose. Для запуска базы данных необходимо выполнить команду *docker compose up*

docker-compose.yaml:

```
version: '3.9'
services:
  mongodb:
    image: mongo:latest
    container_name: mongodb
    restart: unless-stopped
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: password
    ports:
      - 27017:27017
    volumes:
- ./database-data:/data/db
```

2. Проверка работоспособности

Для работы с базой данных будем использовать инструмент *mongosh.* После выполнения команды *mongosh* мы можем увидеть, что мы успешно подключились к базе данных. Также в логах нашей базы данных можно увидеть, что успешно было установлено подключение

```
> mongosh
Current Mongosh Log ID: 64775780865572fb0940b6d9
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.9.1
Using MongoDB:          6.0.6
Using Mongosh:          1.9.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
```

Рисунок 1 - Вывод после вызова команды "mongosh"


```
mongodb        | {"t":{"$date":"2023-05-31T14:24:12.402+00:00"},"s":"I",  "c":"NETWORK",  "id":22943,   "ctx":"listener","msg":"Connection accepted","attr":{"
remote":"172.20.0.1:57798","uuid":"3448c956-4c3d-47d7-ab6b-adbf8f5c6922","connectionId":2,"connectionCount":2}}
mongodb        | {"t":{"$date":"2023-05-31T14:24:12.404+00:00"},"s":"I",  "c":"NETWORK",  "id":51800,   "ctx":"conn2","msg":"client metadata","attr":{"remote"
:"172.20.0.1:57798","client":"conn2","doc":{"application":{"name":"mongosh 1.9.1"},"driver":{"name":"nodejs|mongosh","version":"5.5.0|1.9.1"},"platform":"Node
.js v16.20.0, LE","os":{"name":"darwin","architecture":"arm64","version":"22.3.0","type":"Darwin"}}}}
```


Рисунок 2 - логи базы данных

3. Выполнение методов

1. db.help()

```
  Database Class:

    getMongo                                Returns the current database connection
    getName                                 Returns the name of the DB
    getCollectionNames                      Returns an array containing the names of all collections in the current database.
    getCollectionInfos                      Returns an array of documents with collection information, i.e. collection name and options, for the current da
tabase.
    runCommand                              Runs an arbitrary command on the database.
    adminCommand                            Runs an arbitrary command against the admin database.
    aggregate                               Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
    getSiblingDB                            Returns another database without modifying the db variable in the shell environment.
    getCollection                           Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
    dropDatabase                            Removes the current database, deleting the associated data files.
    createUser                              Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error
if the user already exists on the database.
    updateUser                              Updates the user's profile on the database on which you run the method. An update to a field completely replace
s the previous field's values. This includes updates to the user's roles array.
    changeUserPassword                      Updates a user's password. Run the method in the database where the user is defined, i.e. the database you crea
ted the user.
    logout                                  Ends the current authentication session. This function has no effect if the current session is not authenticate
d.
    dropUser                                Removes the user from the current database.
    dropAllUsers                            Removes all users from the current database.
    auth                                    Allows a user to authenticate to the database from within the shell.
    grantRolesToUser                        Grants additional roles to a user.
    revokeRolesFromUser                     Removes a one or more roles from a user on the current database.
    getUser                                 Returns user information for a specified user. Run this method on the user's database. The user must exist on t
he database on which the method runs.
    getUsers                                Returns information for all the users in the database.
    createCollection                        Create new collection
    createEncryptedCollection               Creates a new collection with a list of encrypted fields each with unique and auto-created data encryption keys
 (DEKs). This is a utility function that internally utilises ClientEnryption.createEncryptedCollection.
    createView                              Create new view
```

Рисунок 3 - вывод команды db.help()

2. db.stats()

Для выполнения данной команды нам необходимо подключиться к базе данных с логином и паролем

Рисунок 4 - подключение с паролем

Рисунок 5 - вывод функции db.stats()

4. Создание новой БД

Для создания новой базы данных воспользуемся функцией *use*

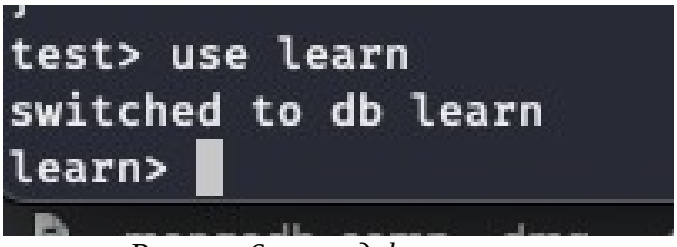

Рисунок 6 - вывод функции use

5. Получение списка доступных БД

Для получения списка доступных баз данных воспользуемся командой админа *listDatabases.* Вместо единицы в качестве значения можно использовать любой поддерживаем тип данных в mongo, так как это значение не влияет на результат выполнения команды.

```
learn> db.adminCommand(
...     {
...       listDatabases: 1
...     }
... )
{
  databases: [
    { name: 'admin', sizeOnDisk: Long("102400"), empty: false },
    { name: 'config', sizeOnDisk: Long("110592"), empty: false },
    { name: 'local', sizeOnDisk: Long("73728"), empty: false }
  ],
  totalSize: Long("286720"),
  totalSizeMb: Long("0"),
  ok: 1
}
```

Рисунок 7 - вывод команды для списка бд

6. Создание новой коллекции и вставка данных

Воспользуемся функцией *db.createCollection(<name>)* для создания коллекции под названием *“unicorns”*. Затем вставим в нее новый документ *{name: 'Aurora', gender: 'f', weight: 450},* для этого воспользуемся функцией *insert* у коллекции *unicorns*

```
learn> db.createCollection("unicorns")
{ ok: 1 }
learn> db.unicorns.insert({name: 'Aurora', gender: 'f', weight: 450})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64775cbdc95545bb73b59b91") }
}
```

Рисунок 8 - создание коллекции и вставка документа

7. Просмотр списка текущих коллекций

Для просмотра списка текущих коллекций мы можем воспользоваться методом *db.runCommand({ listCollections: 1 })*

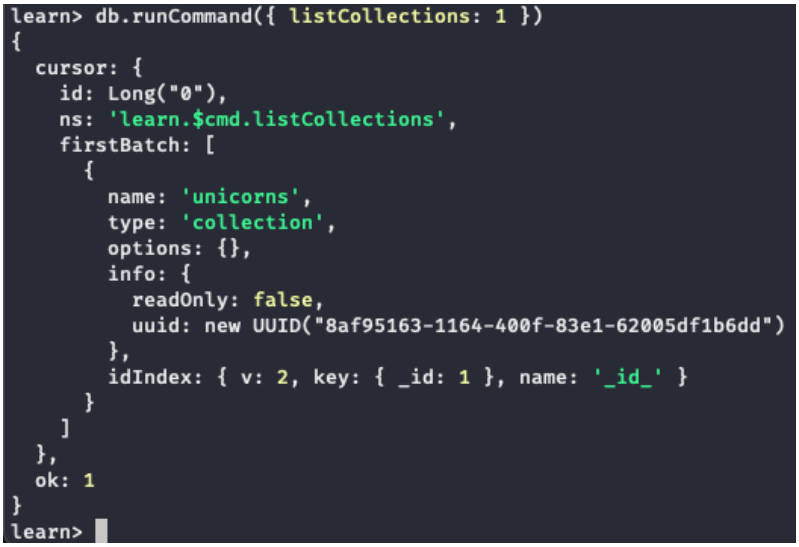

Рисунок 9 - список текущих коллекций

8. Переименование коллекции

Для переименования коллекции *unicorns* воспользуемся методом *db.collection_name.renameCollection(<newName>)*

Рисунок 10 - переименование коллекции unicorns

9. Статистика коллекции

Для просмотра стастики коллекции *unicorns_new* воспользуемся функцией *db.unicorns_new.stats()*

Рисунок 11 - статистика коллекции unicorns_new

10. Удаление коллекции и БД

Для удаления коллекции воспользуемся методом *db.unicorns_new.drop()*. Для удаления базы данных воспользуемся методом *db.dropDatabase()*

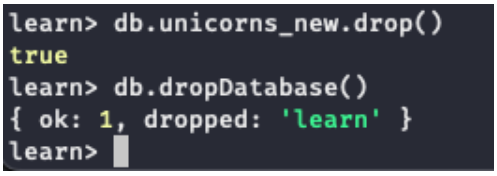

Рисунок 12 - результаты удаления

11. Вывод

В ходе выполнения данной лабораторной работы была установлена база данных mongodb, инструмент для работы с базой данных mongosh и были выполнены различные функции в базе. В итоге удалось овладеть навыками установки mongodb