Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №6 «ВВЕДЕНИЕ В СУБД MONGODB. УСТАНОВКА MONGODB. НАЧАЛО РАБОТЫ С БД»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Трубников А.П

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.

Санкт-Петербург 2024

Цель работы:

овладеть практическими навыками установки СУБД MongoDB..

Практическое задание:

1. Установите MongoDB для обеих типов систем (32/64 бита).
2. Проверьте работоспособность системы запуском клиента mongo.
3. Выполните методы:

db.help()

db.help

db.stats()

4. Создайте БД learn.
5. Получите список доступных БД.
6. Создайте коллекцию unicorns, вставив в нее документ {name: 'Aurora', gender: 'f', weight: 450}.
7. Просмотрите список текущих коллекций.
8. Переименуйте коллекцию unicorns.
9. Просмотрите статистику коллекции.
10. Удалите коллекцию.
11. Удалите БД learn.

Задание 3

Команда db.help()

```
> 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 database.
  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 replaces 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 created the user.
  logout                    Ends the current authentication session. This function has no effect if the current session is not authenticated.
  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 the database on which
                            the method runs.
  getUsers                  Returns information for all the users in the database.
```

Команда db.help

```
>_MONGOSH
> 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 database.
  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 replaces 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 created the user.
  logout                    Ends the current authentication session. This function has no effect if the current session is not authenticated.
  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 the database on which
                            the method runs.
  getUsers                  Returns information for all the users in the database.
```

Команда db.stats()

```
> db.stats()
< {
    db: 'test',
    collections: Long('0'),
    views: Long('0'),
    objects: Long('0'),
    avgObjSize: 0,
    dataSize: 0,
    storageSize: 0,
    indexes: Long('0'),
    indexSize: 0,
    totalSize: 0,
    scaleFactor: Long('1'),
    fsUsedSize: 0,
    fsTotalSize: 0,
    ok: 1
  }
test>
```

Задание 4

Создание базы данных:

```
> use <Lab6_BD>
< switched to db <Lab6_BD>
<Lab6_BD>
```

Задание 5

Получите список доступных БД.:

```
> show dbs
< Lab6_BD   40.00 KiB
  admin     40.00 KiB
  config    96.00 KiB
  local     40.00 KiB
```

Задание 6:

Создайте коллекцию unicorns, вставив в нее документ {name: 'Aurora', gender: 'f', weight: 450}.

```
> db.createCollection("unicorns")
< { ok: 1 }
> db.unicorns.insertOne({name: 'Aurora', gender: 'f', weight: 450})
< {
    acknowledged: true,
    insertedId: ObjectId('65e1987c8cb7b7b6bbcdaffe')
  }
```

Задание 7:

Просмотрите список текущих коллекций.

```
> show collections
< unicorns
```

Задание 8:

Переименуйте коллекцию unicorns.

```
> db.unicorns.renameCollection("pages")
< { ok: 1 }
```

Задание 9:

Просмотрите статистику коллекции:

```
> db.pages.stats()
< {
    ok: 1,
    capped: false,
    wiredTiger: {
      metadata: { formatVersion: 1 },
      creationString: 'access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=(commit_timestamp=none,durable_timestamp=none,read_timestamp=none,write_
      type: 'file',
      uri: 'statistics:table:collection-7-4160459546396688383',
      LSM: {
        'bloom filter false positives': 0,
        'bloom filter hits': 0,
        'bloom filter misses': 0,
        'bloom filter pages evicted from cache': 0,
        'bloom filter pages read into cache': 0,
        'bloom filters in the LSM tree': 0,
        'chunks in the LSM tree': 0,
        'highest merge generation in the LSM tree': 0,
        'queries that could have benefited from a Bloom filter that did not exist': 0,
```

```
    }
  },
  sharded: false,
  size: 65,
  count: 1,
  numOrphanDocs: 0,
  storageSize: 20480,
  totalIndexSize: 20480,
  totalSize: 40960,
  indexSizes: { _id_: 20480 },
  avgObjSize: 65,
  ns: 'Lab6_BD.pages',
  nindexes: 1,
  scaleFactor: 1
```

Задание 10:

Удалите коллекцию

```
> db.pages.drop()
< true
```

Задание 11:

Удалите БД learn.

```
> db.dropDatabase()
< { ok: 1, dropped: 'Lab6_BD' }
```

Вывод:

В ходе выполнения практического задания по установке СУБД MongoDB были изучены основные шаги по установке и настройке сервера базы данных.