

Отчет по лабораторной работе 7, вариант 7.

Выполнил: Золотухин Андрей Александрович КС-36

Преподаватель: Семёнов Геннадий Николаевич

ЗАДАНИЕ

Практическое задание:

1. Установите MongoDB в зависимости от типов систем (32/64 бита).
2. Проверьте работоспособность системы запуском клиента **mongo** или **mongosh**.
3. Выполните методы:
 - a) `db.help()`
 - b) `db.help`
 - c) `db.stats()`
4. Получите список доступных БД.
5. Создайте БД в соответствии с вариантом задания Лабораторной работы №1, имя БД вводится латиницей в формате «Фамилия_группа», например,
`db> use Ivanov_ks30`
6. Создайте коллекцию, вставив в нее документы в соответствии со строками таблицы вашего варианта задания Лабораторной работы №1

Например:

Результаты ЕГЭ:

Дата экзамена	Школа, №	Балл	Ученик	Возраст	Дисциплина	Экзаменатор
3.06.06	114	85	Иванов	18	Математика	Кузнецов
15.06.06	295	79	Петрова	17	Русский Язык	Филатова
2.06.06	1197	91	Сидорова	19	Математика	Кузнецов
2.06.06	114	62	Иванов	18	Русский Язык	Филатова
14.06.06	1197	77	Сидорова	19	Физика	Гришин
4.06.06	295	81	Петрова	17	Математика	Кузнецов

В СУБД MongoDB создайте базу данных

7. Выведите список текущих коллекций вашей БД.
8. Выведите содержимое созданной коллекции

9. В приложении **MongoDB Compass** сделайте экспорт коллекции вашей БД в файл *.json.

10. Файл отчета в формате WORD и файл коллекции в формате *.json загрузите на учебный портал в курса «Управление данными» в раздел «Сдать Лаб.7»

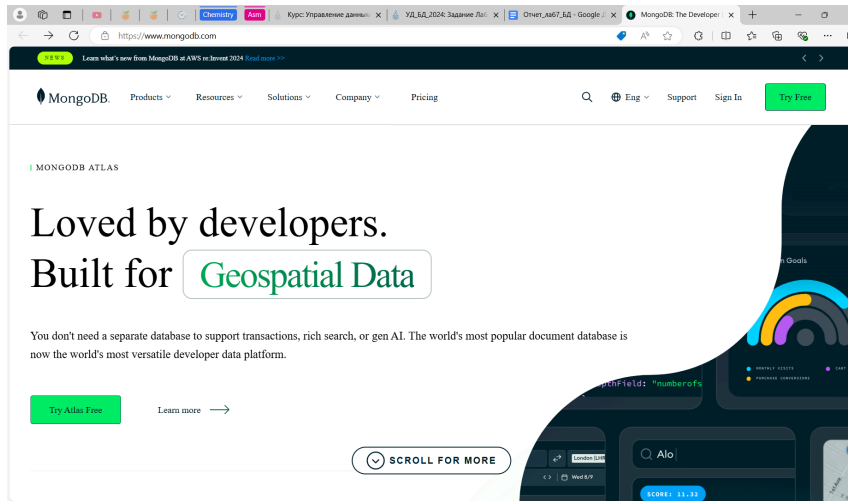
Цель работы

Научиться практическим навыкам установки MongoDB, созданием базы данных и коллекций.

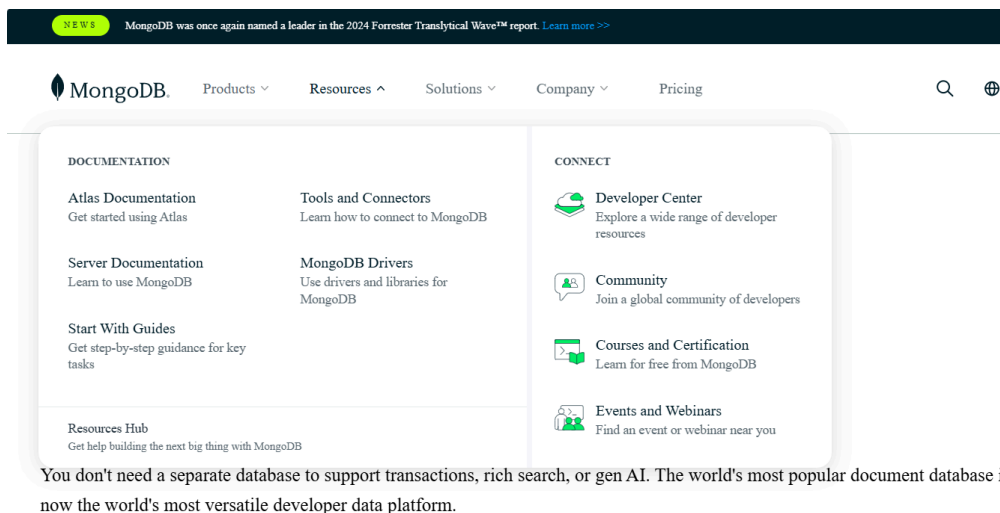
Выполнение задания

1) Установите MongoDB в зависимости от типов систем (32/64 бита):

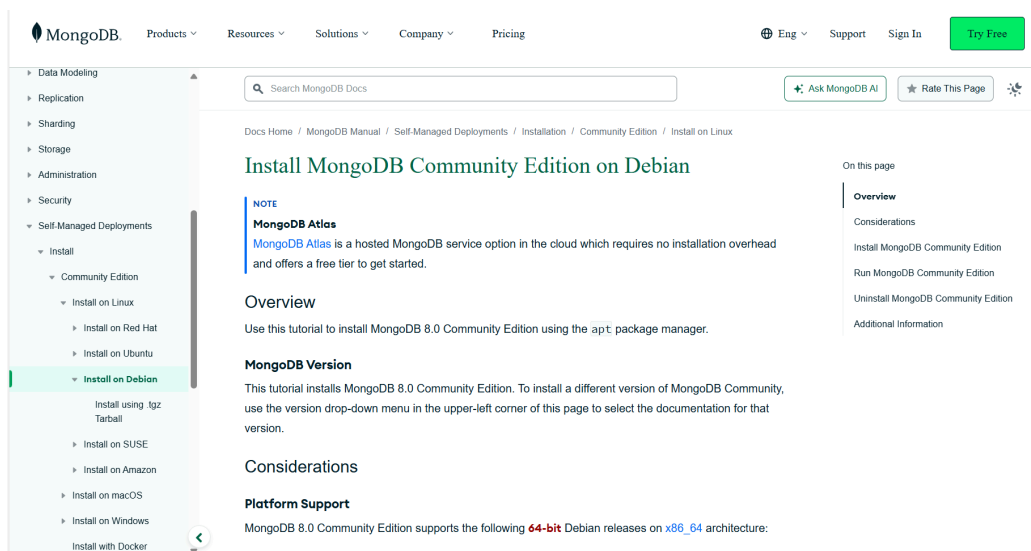
Перехожу на главную сайт mongodb:



Нажимаю на вкладку *Resources* и перехожу на страницу *Server Documentation*:



Перехожу по пути *Self-Managed Deployments* => *Install* => *Community Edition* => *Install on Linux* => *Install on Debian*



Чтобы установить MongoDB Community Edition, следую данным шагам:

1) Импортируйте открытый ключ.

a) Установите `gnupg` и `curl` с помощью команды:

```
sudo apt-get install gnupg curl
```

b) Чтобы импортировать открытый ключ:

```
curl -fsSL  
https://www.mongodb.org/static/pgp/server-8.0.asc |  
\  
sudo gpg -o  
/usr/share/keyrings/mongodb-server-8.0.gpg \  
--dearmor
```

2) Создайте файл списка.

```
echo "deb [  
signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ]  
http://repo.mongodb.org/apt/debian  
bookworm/mongodb-org/8.0 main" | sudo tee  
/etc/apt/sources.list.d/mongodb-org-8.0.list
```

3) Перезагрузите базу данных пакетов.

```
sudo apt-get update
```

4) Установите MongoDB Community Server.

```
sudo apt-get install -y mongodb-org
```

2) Проверьте работоспособность системы запуском клиента *mongo* или *mongosh*:

```
corgi@LAPTOP-AS2LVH3V:~$ mongosh
Current Mongosh Log ID: 675fdab7da8a690e4da5c0ac
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.6
Using MongoDB:      8.0.4
Using Mongosh:       2.3.6

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

-----
The server generated these startup warnings when booting
2024-12-16T07:14:25.839+03:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-12-16T07:14:26.104+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-12-16T07:14:26.104+03:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2024-12-16T07:14:26.104+03:00: We suggest setting the contents of sysfsFile to 0.
2024-12-16T07:14:26.104+03:00: vm.max_map_count is too low
2024-12-16T07:14:26.104+03:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
-----
test> █
```

3) Выполните методы:

a) db.help()

```
test> 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 database 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
```

b) db.help

```
test> 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 database 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
```

c) db.stats()

```
test> 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
}
```

4) Получите список доступных БД.

```
test> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
```

5) Создайте БД в соответствии с вариантами задания
Лабораторной работы №1, имя БД вводится латиницей в
формате “Фамилия_группа”

```
test> use Zolotukhin_ks36
switched to db Zolotukhin_ks36
Zolotukhin_ks36> █
```

- 6) Создайте коллекцию, вставив в нее документы в соответствии со строками таблицы вашего варианта задания Лабораторной работы №1.

```
Zolotukhin_ks36> db.dormitory.insertOne({"accommodation_date": "3.08.05", "group_name": "KC-10", "student_name": "Васильев", "neighbour_name": "Гришин", "distance": 200, "room_number": 14, "scholarship": 2000});
{
  acknowledged: true,
  insertedId: ObjectId('675fed7eda8a690e4da5c0b9')
}
Zolotukhin_ks36> db.dormitory.insertOne({"accommodation_date": "15.08.05", "group_name": "KC-14", "student_name": "Петрова", "neighbour_name": "Сидорова", "distance": 435, "room_number": 37, "scholarship": 2570});
{
  acknowledged: true,
  insertedId: ObjectId('675fedc8da8a690e4da5c0ba')
}
Zolotukhin_ks36> db.dormitory.insertOne({"accommodation_date": "2.08.05", "group_name": "KC-14", "student_name": "Сидоров", "neighbour_name": "Иванов", "distance": 112, "room_number": 25, "scholarship": 2000});
{
  acknowledged: true,
  insertedId: ObjectId('675fede9da8a690e4da5c0bb')
}
Zolotukhin_ks36> db.dormitory.insertOne({"accommodation_date": "2.08.05", "group_name": "KC-10", "student_name": "Иванов", "neighbour_name": "Сидоров", "distance": 240, "room_number": 25, "scholarship": 2240});
{
  acknowledged: true,
  insertedId: ObjectId('675fee09da8a690e4da5c0bc')
}
Zolotukhin_ks36> db.dormitory.insertOne({"accommodation_date": "14.08.05", "group_name": "KC-10", "student_name": "Сидорова", "neighbour_name": "Петрова", "distance": 1200, "room_number": 37, "scholarship": 4500});
{
  acknowledged: true,
  insertedId: ObjectId('675fee2eda8a690e4da5c0bd')
}
Zolotukhin_ks36> db.dormitory.insertOne({"accommodation_date": "4.08.05", "group_name": "KC-14", "student_name": "Гришин", "neighbour_name": "Васильев", "distance": 780, "room_number": 14, "scholarship": 4000});
{
  acknowledged: true,
  insertedId: ObjectId('675fee4fda8a690e4da5c0be')
}
Zolotukhin_ks36> █
```

- 7) Выведите список текущих коллекций вашей БД.

```
Zolotukhin_ks36> show collections
dormitory
Zolotukhin_ks36> █
```

- 8) Выведите содержимое созданной коллекции.

```
Zolotukhin_ks36> db.dormitory.find()
[
  {
    _id: ObjectId('675fed7eda8a690e4da5c0b9'),
    accommodation_date: '3.08.05',
    group_name: 'КС-10',
    student_name: 'Васильев',
    neighbour_name: 'Гришин',
    distance: 200,
    room_number: 14,
    scholarship: 2000
  },
  {
    _id: ObjectId('675fedc8da8a690e4da5c0ba'),
    accommodation_date: '15.08.05',
    group_name: 'КС-14',
    student_name: 'Петрова',
    neighbour_name: 'Сидорова',
    distance: 435,
    room_number: 37,
    scholarship: 2570
  },
  {
    _id: ObjectId('675fede9da8a690e4da5c0bb'),
    accommodation_date: '2.08.05',
    group_name: 'КС-14',
    student_name: 'Сидоров',
    neighbour_name: 'Иванов',
    distance: 112,
    room_number: 25,
    scholarship: 2000
  },
  {
    _id: ObjectId('675fee09da8a690e4da5c0bc'),
    accommodation_date: '2.08.05',
    group_name: 'КС-10',
    student_name: 'Иванов',
    neighbour_name: 'Сидоров',
    distance: 240,
    room_number: 25,
    scholarship: 2240
  },
  {
    _id: ObjectId('675fee2eda8a690e4da5c0bd'),
    accommodation_date: '14.08.05',
    group_name: 'КС-10',
    student_name: 'Сидорова',
    neighbour_name: 'Петрова',
    distance: 1200,
    room_number: 37,
    scholarship: 4500
  },
  {
    _id: ObjectId('675fee4fda8a690e4da5c0be'),
    accommodation_date: '4.08.05',
    group_name: 'КС-14',
    student_name: 'Гришин',
    neighbour_name: 'Васильев',
    distance: 780,
    room_number: 14,
    scholarship: 4000
  }
]
Zolotukhin_ks36> █
```

9) Сделайте экспорт коллекции вашей БД в файл *.json.

```
mongoexport --db Zolotukhin_ks36 --collection dormitory --out  
lab7.json
```

```
corgi@LAPTOP-AS2LVH3V:~/git_repositories/DataManagement_labs/lab7$ mongoexport --db Zolotukhin_ks36 --  
-collection dormitory --out lab7.json  
2024-12-16T12:25:46.321+0300    connected to: mongodb://localhost/  
2024-12-16T12:25:46.324+0300    exported 6 records  
corgi@LAPTOP-AS2LVH3V:~/git_repositories/DataManagement_labs/lab7$
```

Выводы

В ходе лабораторной работы я научился управлять базой данных MongoDB, а именно создавать и удалять базу данных, создавать и удалять, экспортировать коллекцию.