

Департамент образования и науки города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики управления и технологий

Мареев Георгий Александрович БД-241м

**Практическая работа 2-1. Часть 1. Изучение методов хранения данных на
основе NoSQL. Вариант 13**

Направление подготовки/специальность
38.04.05 - Бизнес-информатика
Бизнес-аналитика и большие данные
(очная форма обучения)

Руководитель дисциплины:
Босенко Т.М., доцент департамента
информатики, управления и технологий,
кандидат технических наук

Москва
2025

Введение

Цель: получить практические навыки работы с документо-ориентированной базой данных MongoDB путем выполнения основных операций по управлению данными.

Основная часть

Запуск Mongo:

Проверка каталога “mongo”

```
.../dba/nonrel/mongo$ ls
```

```
docker-compose.yml express-app
```

Запуск контейнеров:

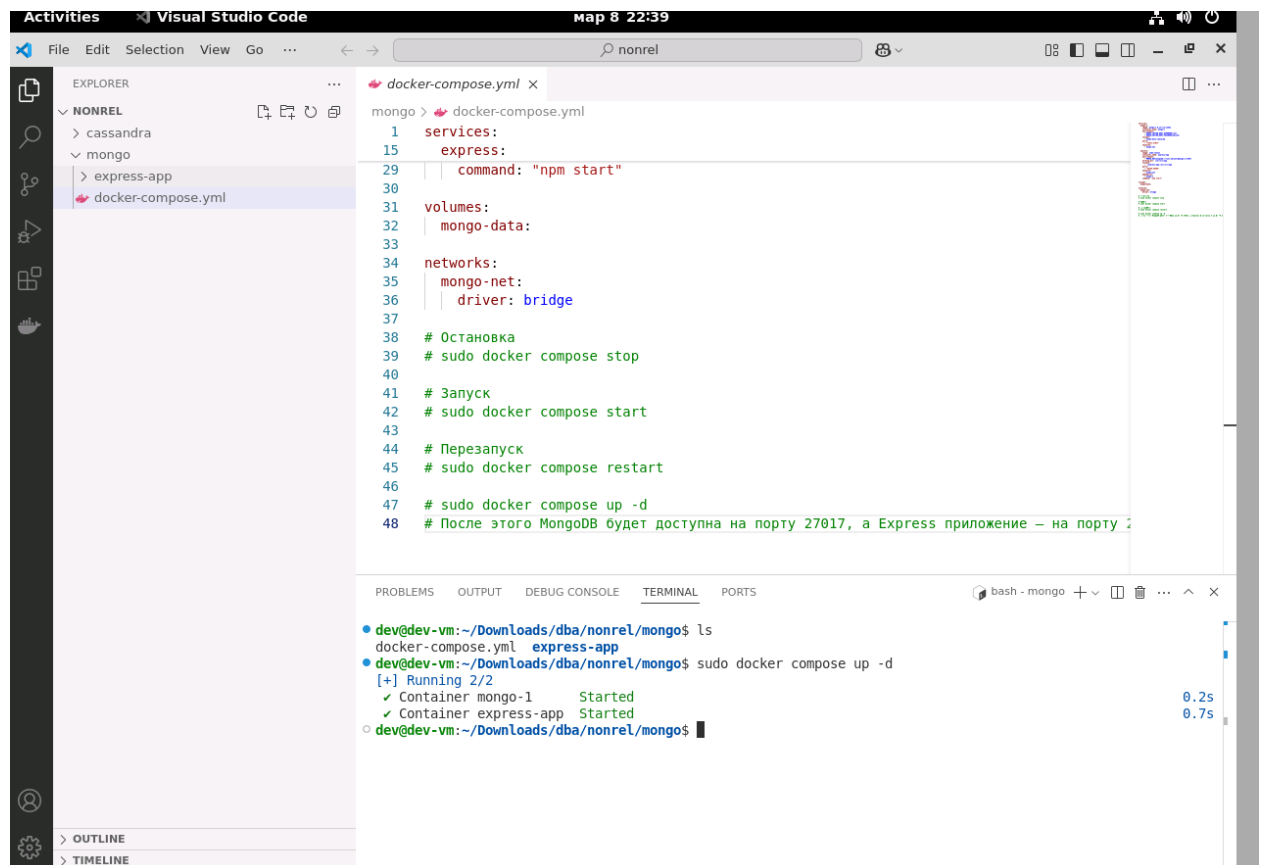
```
sudo docker compose up -d
```

[+] Running 2/2

✓ Container mongo-1 Started

0.2s

✓ Container express-app Started



Нет Network mongo_mongo-net,

Проверка подключения к mongo:

```
sudo docker exec -ti mongo-1 mongosh -u "root" -p "abc123!"
```

Error response from daemon: container

dfd339d83cd2473d643527fa87443e6999e639f150c797b5a38a8e17bd0cd403 is not running

Проверка логов mongo-1:

`sudo docker logs mongo-1`

about to fork child process, waiting until server is ready for connections.

forked process: 26

Ошибка:

WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!

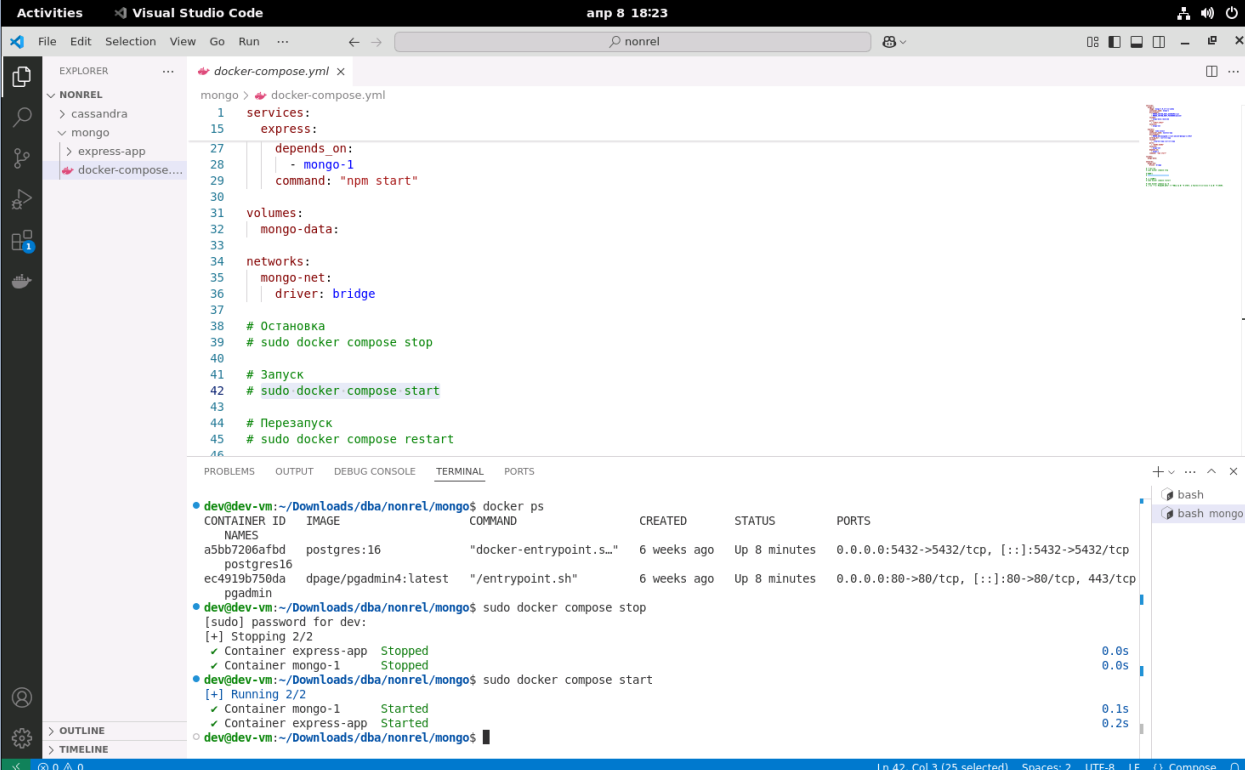
see <https://jira.mongodb.org/browse/SERVER-54407>

see also <https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2>

see also <https://github.com/docker-library/mongo/issues/485#issuecomment-891991814>

Попытка решения проблемы

Предоставленный образ был развернут на другой машине. Результат:



The screenshot shows the Visual Studio Code interface with a Docker Compose file open in the editor and a terminal window at the bottom. The Docker Compose file defines a service named 'express' that depends on 'mongo-1' and runs 'npm start'. The terminal shows the output of 'docker ps' and 'docker compose stop/start' commands, indicating that the containers were successfully started.

```
docker-compose.yml
1 services:
15   express:
27     depends_on:
28       - mongo-1
29     command: "npm start"
30
31 volumes:
32   mongo-data:
33
34 networks:
35   mongo-net:
36     driver: bridge
37
38 # Остановка
39 # sudo docker compose stop
40
41 # Запуск
42 # sudo docker compose start
43
44 # Перезапуск
45 # sudo docker compose restart
46
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
a5bb7206afbd	postgres:16	"docker-entrypoint.s..."	6 weeks ago	Up 8 minutes	0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp
ec4919b750da	dpag/pgadmin4:latest	"/entrypoint.sh"	6 weeks ago	Up 8 minutes	0.0.0.0:80->80/tcp, [::]:80->80/tcp, 443/tcp

```
dev@dev-vm:~/Downloads/dba/nonrel/mongo$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
a5bb7206afbd   postgres:16                        "docker-entrypoint.s..." 6 weeks ago   Up 8 minutes   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp
ec4919b750da   dpag/pgadmin4:latest              "/entrypoint.sh"         6 weeks ago   Up 8 minutes   0.0.0.0:80->80/tcp, [::]:80->80/tcp, 443/tcp

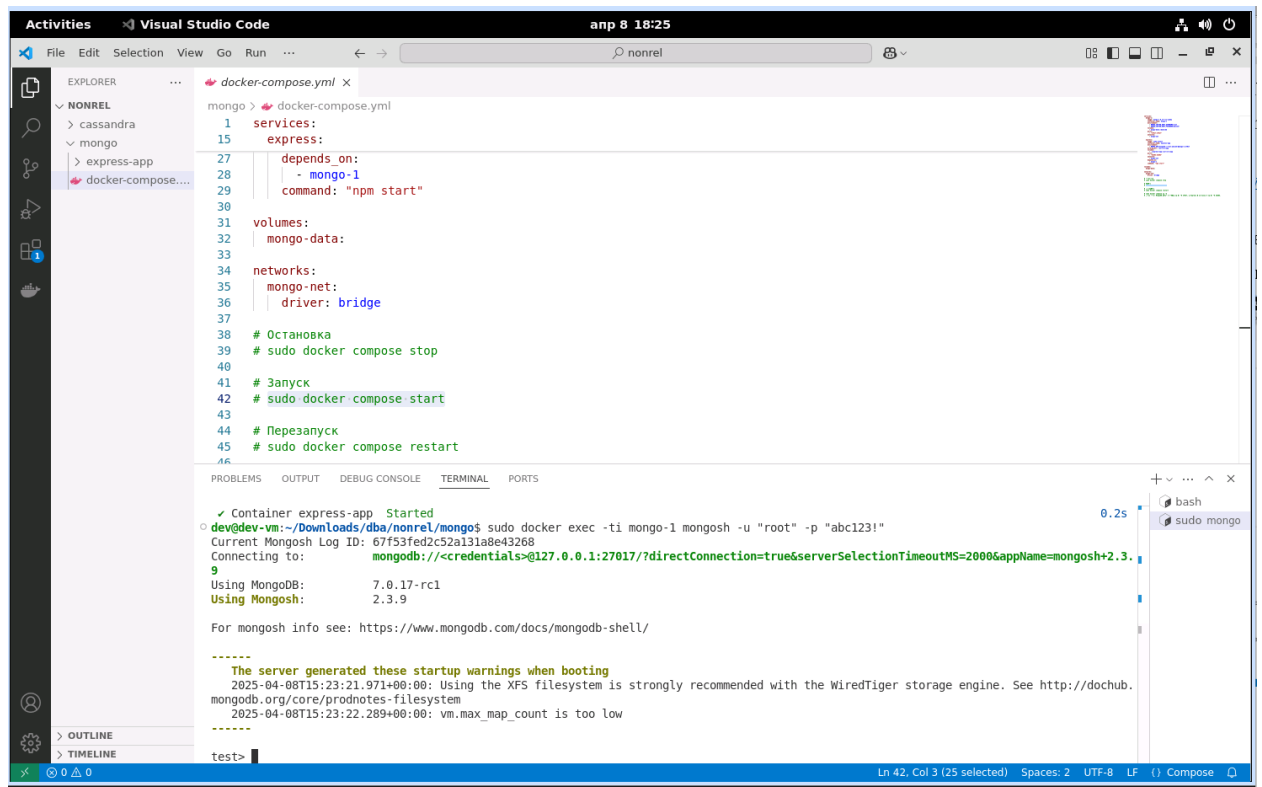
dev@dev-vm:~/Downloads/dba/nonrel/mongo$ sudo docker compose stop
[sudo] password for dev:
[+] Stopping 2/2
 ✓ Container express-app Stopped      0.0s
 ✓ Container mongo-1 Stopped          0.0s

dev@dev-vm:~/Downloads/dba/nonrel/mongo$ sudo docker compose start
[+] Running 2/2
 ✓ Container mongo-1 Started          0.1s
 ✓ Container express-app Started     0.2s

dev@dev-vm:~/Downloads/dba/nonrel/mongo$
```

Попытка подключения:

```
sudo docker exec -ti mongo-1 mongosh -u "root" -p "abc123!"
```



```
docker-compose.yml
1 services:
15 express:
27   depends_on:
28     - mongo-1
29   command: "npm start"
30
31 volumes:
32   mongo-data:
33
34 networks:
35   mongo-net:
36     driver: bridge
37
38 # Остановка
39 # sudo docker compose stop
40
41 # Запуск
42 # sudo docker compose start
43
44 # Перезапуск
45 # sudo docker compose restart
46

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

dev@dev-vm:~/Downloads/dba/nonrel/mongo$ sudo docker exec -ti mongo-1 mongosh -u "root" -p "abc123!"
Current Mongosh Log ID: 67f53fed2c52a131a8e43268
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.
Using MongoDB:      7.0.17-rc1
Using Mongosh:      2.3.9

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

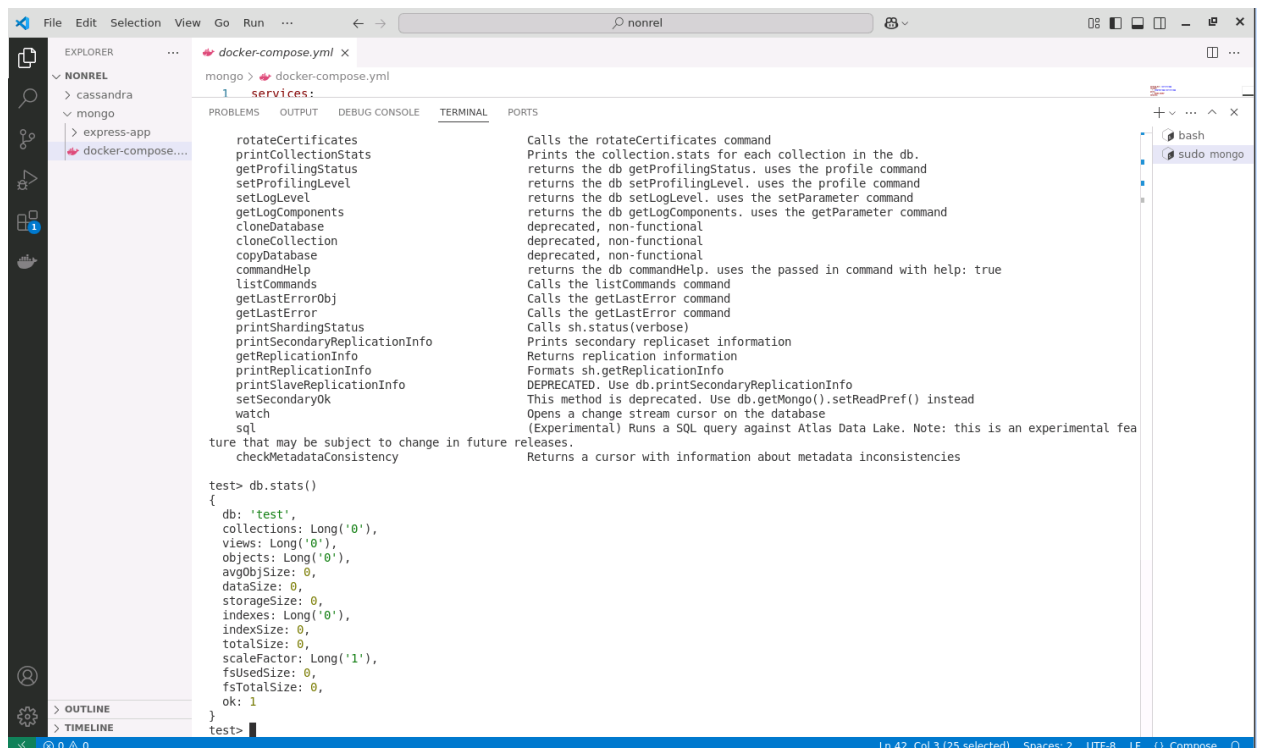
-----
The server generated these startup warnings when booting
2025-04-08T15:23:21.971+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2025-04-08T15:23:22.289+00:00: vm.max_map_count is too low
-----

test>
```

Ура! Работает!

```
db.help()
```

```
db.stats()
```



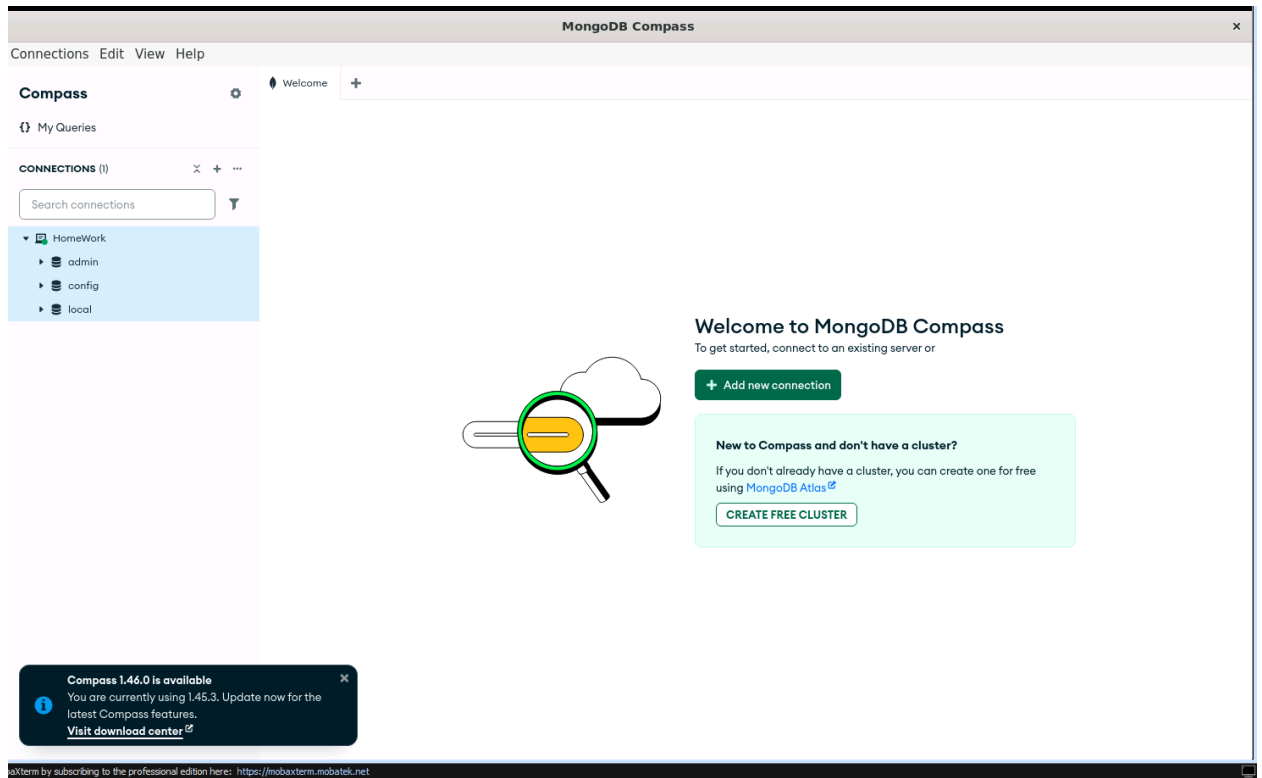
```
rotateCertificates      Calls the rotateCertificates command
printCollectionStats    Prints the collection.stats for each collection in the db.
getProfilingStatus      returns the db getProfilingStatus. uses the profile command
setProfilingLevel       returns the db setProfilingLevel. uses the profile command
setLogLevel             returns the db setLogLevel. uses the setParameter command
getLogComponents        returns the db getLogComponents. uses the getParameter command
cloneDatabase            deprecated, non-functional
cloneCollection          deprecated, non-functional
copyDatabase             deprecated, non-functional
commandHelp             returns the db commandHelp. uses the passed in command with help: true
listCommands            Calls the listCommands command
getLastErrorObj          Calls the getLastError command
getLastError             Calls the getLastError command
printShardingStatus      Calls sh.status(verbose)
printSecondaryReplicationInfo Prints secondary replicaset information
getReplicationInfo       Returns replication information
printReplicationInfo     Formats sh.getReplicationInfo
printSlaveReplicationInfo DEPRECATED. Use db.printSecondaryReplicationInfo
setSecondaryOk           This method is deprecated. Use db.getMongo().setReadPref() instead
watch                   Opens a change stream cursor on the database
sql                     (Experimental) Runs a SQL query against Atlas Data Lake. Note: this is an experimental feature that may be subject to change in future releases.
checkMetadataConsistency Returns a cursor with information about metadata inconsistencies

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
}

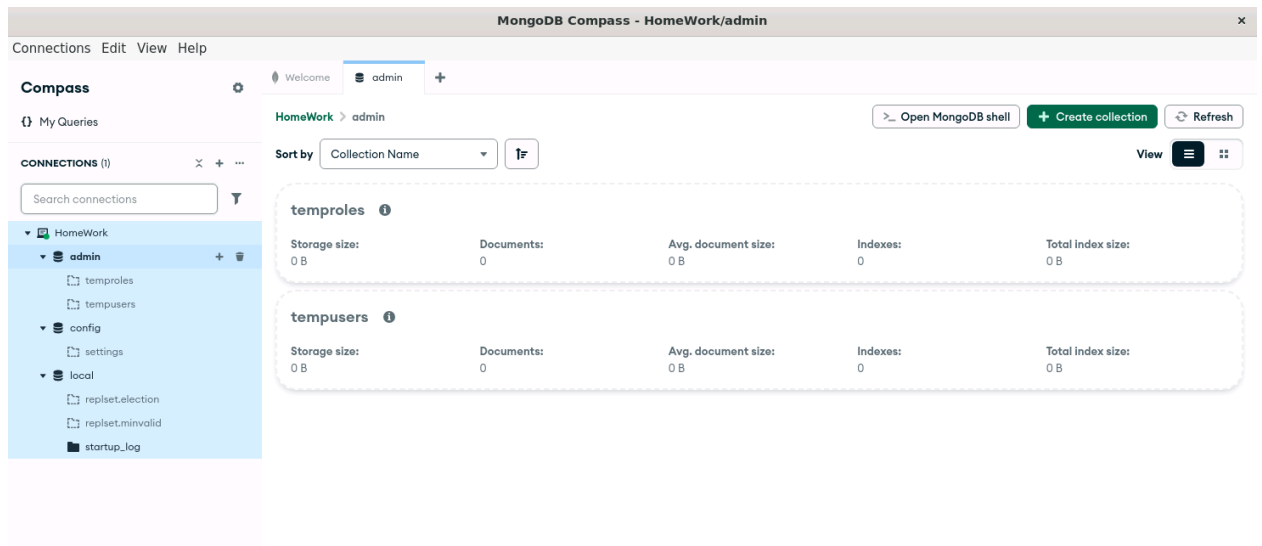
test>
```

Использование настольных приложений

MongoDB Compass

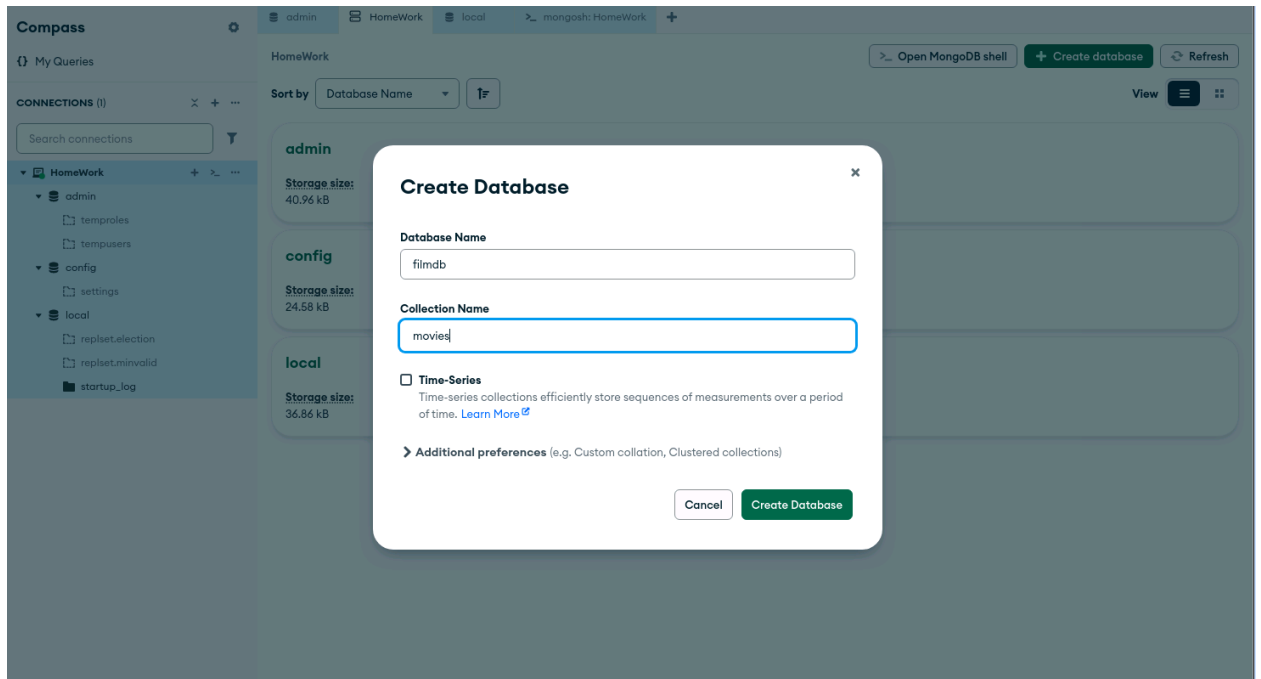


Подключение HomeWork

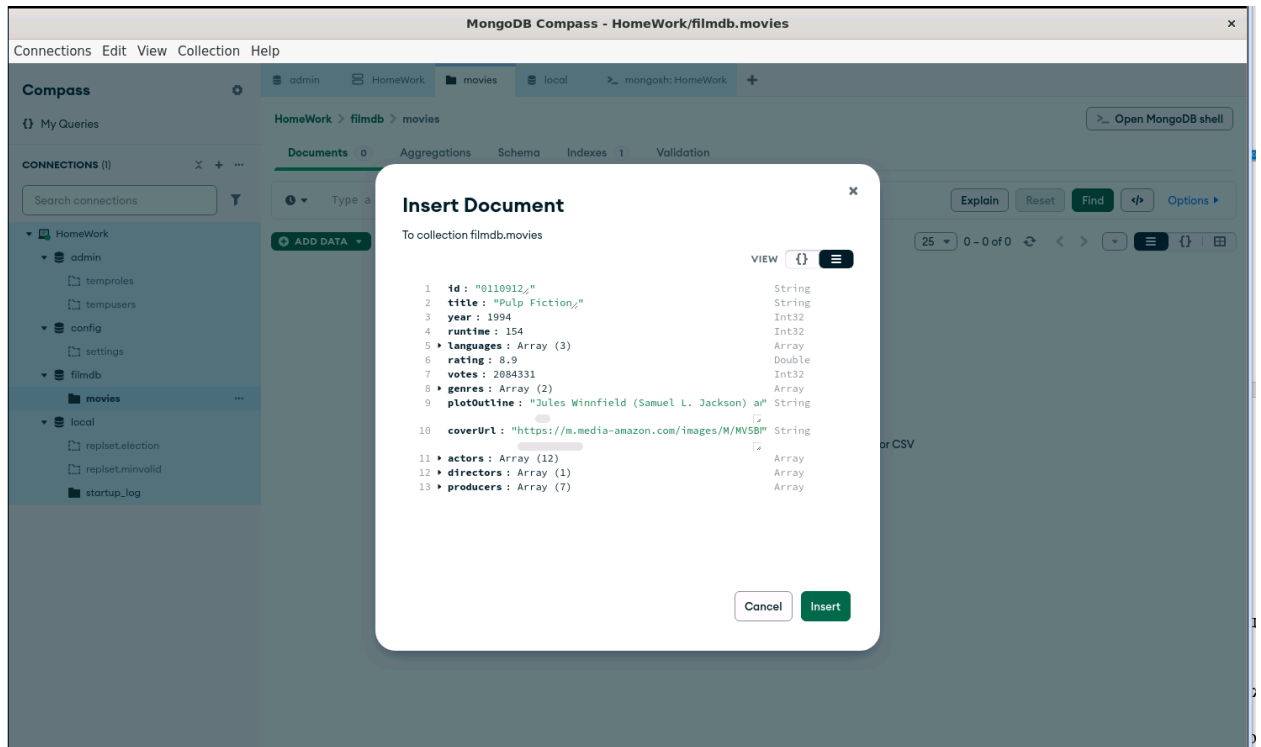


Работа в MongoDB

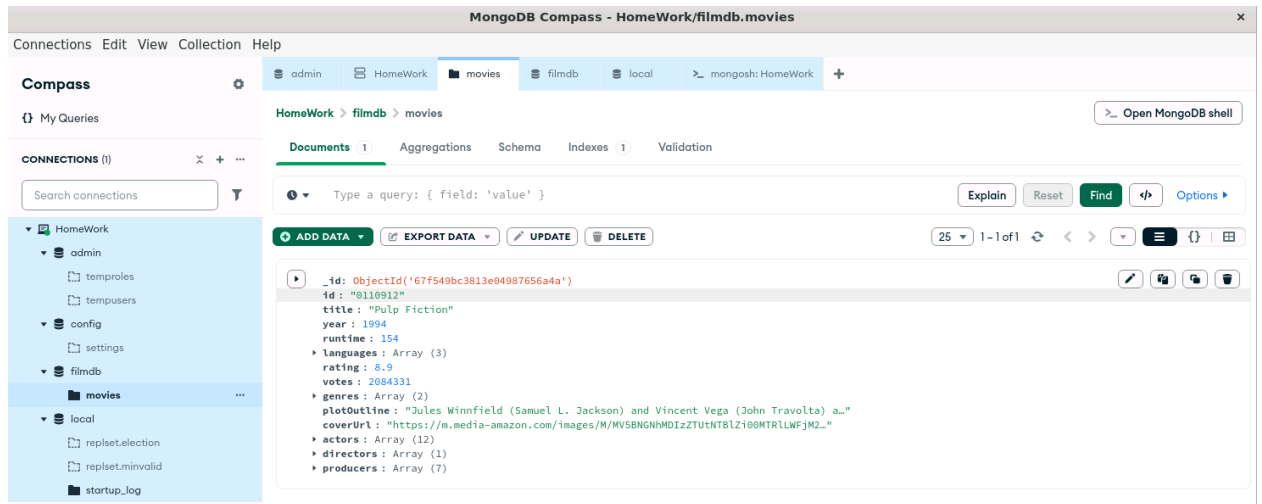
Создание новой базы данных (filmdb)



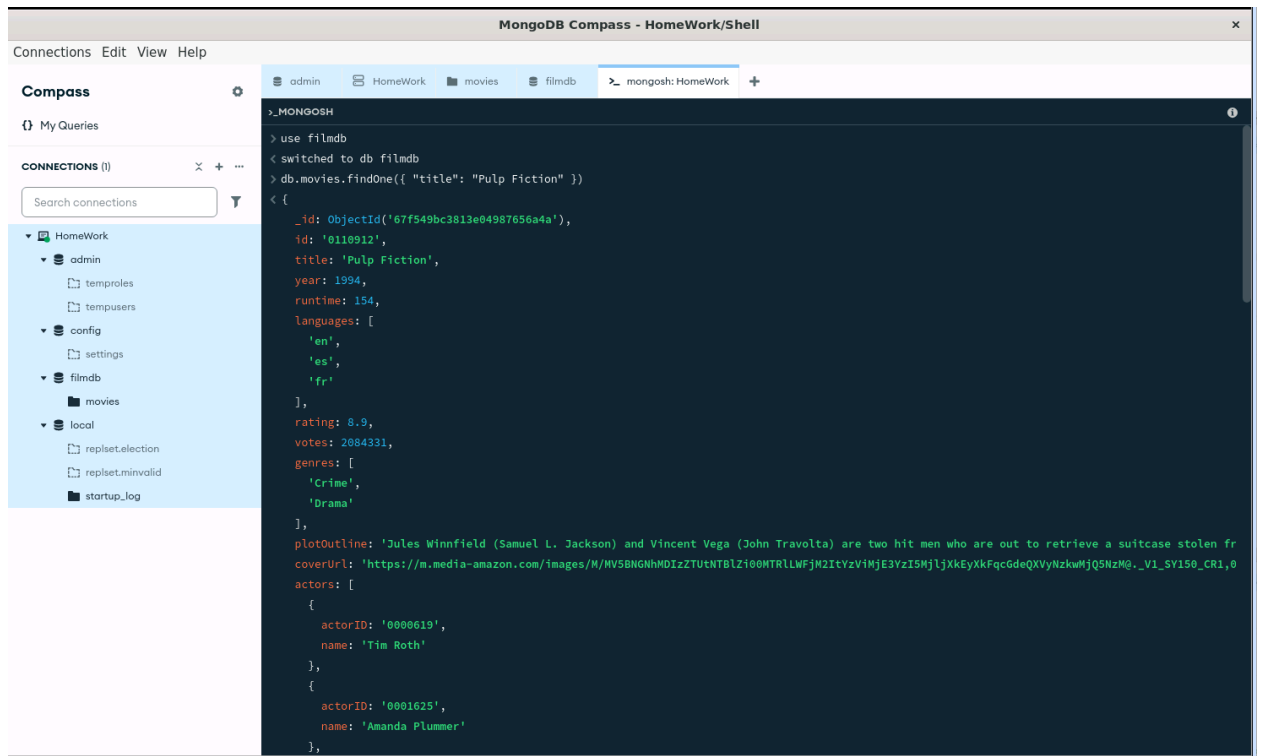
Добавление данных



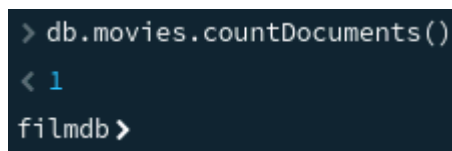
Проверка вставки



```
db.movies.findOne({ "title": "Pulp Fiction" })
```

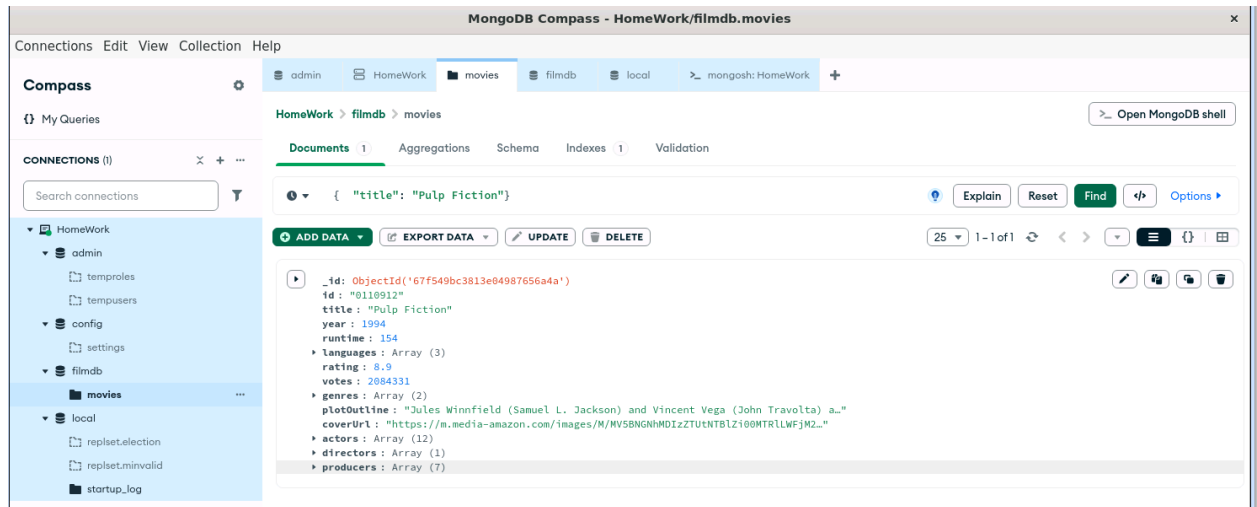


```
db.movies.countDocuments()
```



Поиск документа

Фильтр: {"title": "Pulp Fiction"}



Вставка фильма «The Matrix»

```
db.movies.insertOne(
```

```
{
```

```
  "id": "0133093",
```

```
  "title": "The Matrix",
```

```
  "year": 1999,
```

```
  "runtime": 136,
```

```
  "languages": ["en"],
```

```
  "rating": 8.7,
```

```
  "votes": 1496538,
```

```
  "genres": ["Action", "Sci-Fi"],
```

```
  "plotOutline": "Thomas A. Anderson is a man living two lives. By day he is an average computer programmer and by night a hacker known as Neo. Neo has always questioned his reality, but the truth is far beyond his imagination. Neo finds himself targeted by the police when he is contacted by Morpheus, a legendary computer hacker branded a terrorist by the government. Morpheus awakens Neo to the real world, a ravaged wasteland where most of humanity have been captured by a race of machines that live off of the humans' body heat and electrochemical energy and who imprison their minds within an artificial reality known as the Matrix. As a rebel against the machines, Neo must return to the Matrix and confront the agents: super-powerful computer programs devoted to snuffing out Neo and the entire human rebellion.",
```

"coverUrl": "https://m.media-amazon.com/images/M/MV5BNzQzOTk3OTAtNDQ0Zi00ZTVkLWI0MTEtMDllZjNkYzNjNTc4L2ltYWdlXkEyXkFqcGdeQXVyNjU0OTQ0OTY@._V1_SX101_CR0,0,101,150_.jpg",

"actors": [

- { "actorID": "0000206", "name": "Keanu Reeves"},
- { "actorID": "0000401", "name": "Laurence Fishburne"},
- { "actorID": "0005251", "name": "Carrie-Anne Moss"},
- { "actorID": "0915989", "name": "Hugo Weaving"},
- { "actorID": "0287825", "name": "Gloria Foster"},
- { "actorID": "0001592", "name": "Joe Pantoliano"},
- { "actorID": "0159059", "name": "Marcus Chong"},
- { "actorID": "0032810", "name": "Julian Arahanga"},
- { "actorID": "0000246", "name": "Bruce Willis"},
- { "actorID": "0000609", "name": "Ving Rhames"},
- { "actorID": "0000235", "name": "Uma Thurman"},
- { "actorID": "0000233", "name": "Quentin Tarantino"}

],

"directors": [

- { "directorID": "0905154", "name": "Lana Wachowski"},
- { "directorID": "0905152", "name": "Lilly Wachowski"}

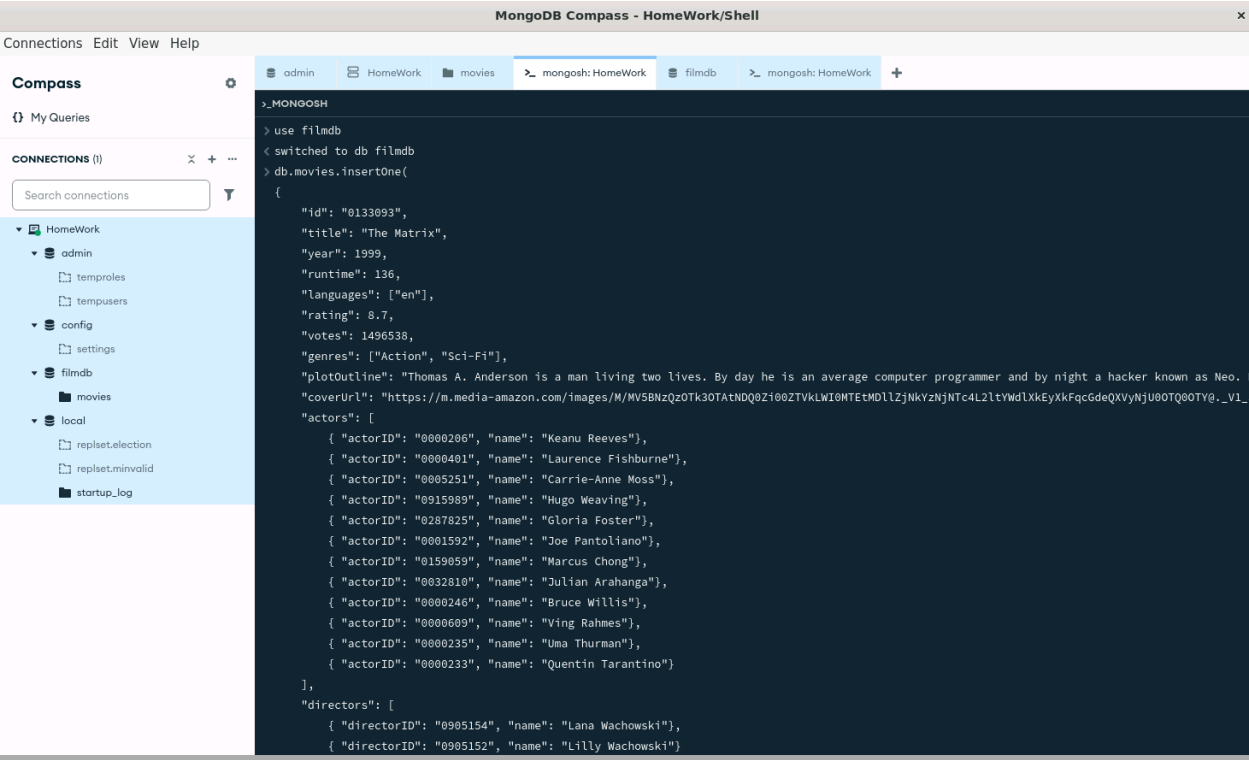
],

"producers": [

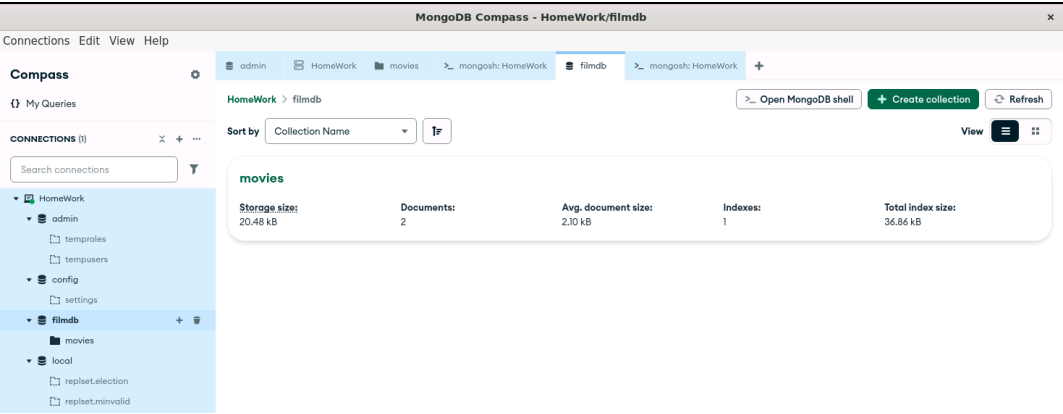
- { "producerID": "0075732", "name": "Bruce Berman"},
- { "producerID": "0185621", "name": "Dan Cracchiolo"},
- { "producerID": "0400492", "name": "Carol Hughes"}

]

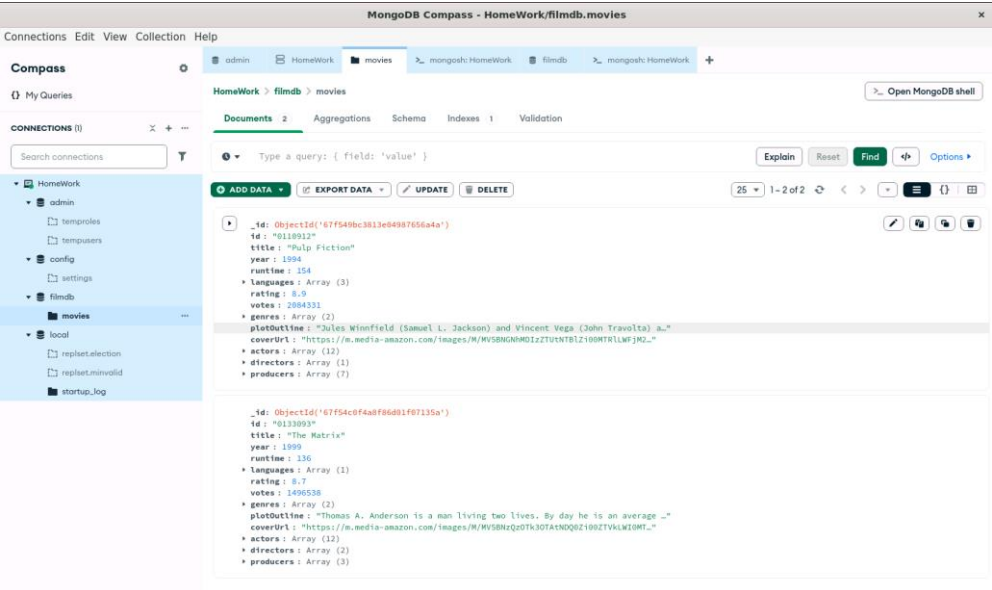
}}



Проверка вставки



Теперь 2 документа



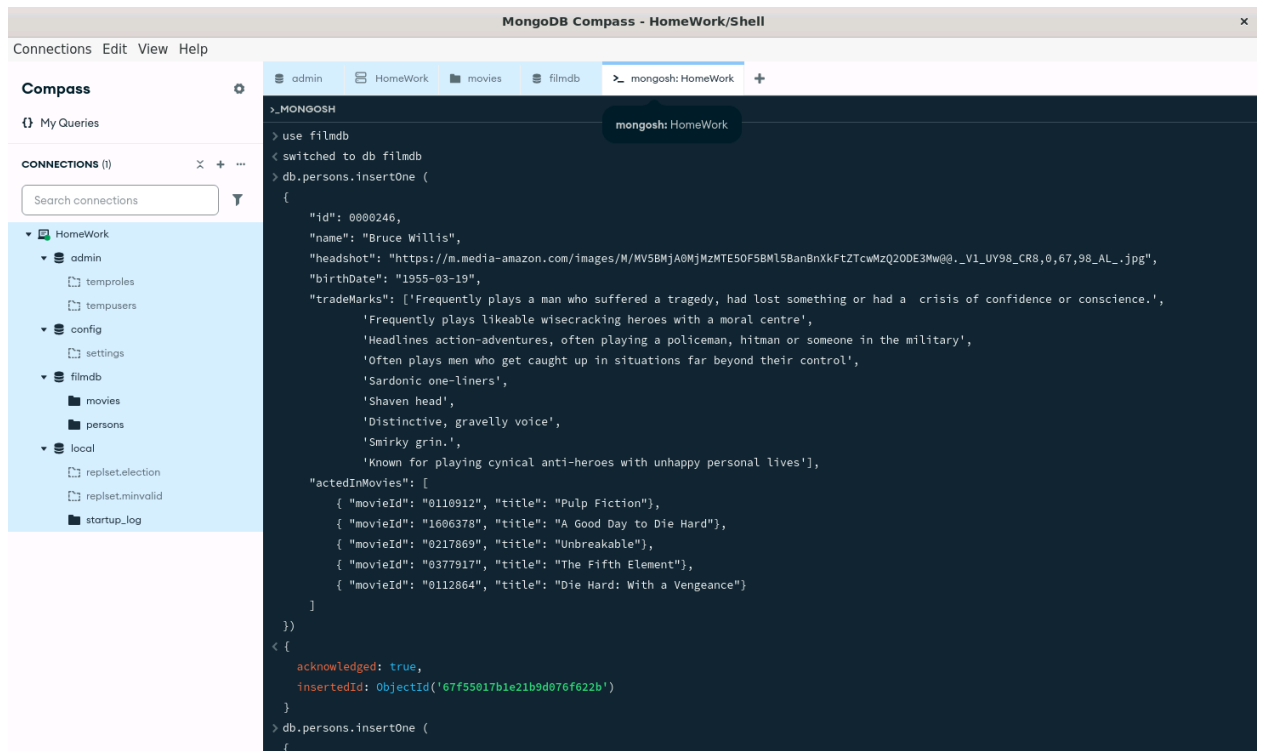
db.getCollectionNames()

```
> db.getCollectionNames()
< [ 'movies' ]
filmdb>
```

db.movies.getIndexes()

```
> db.movies.getIndexes()
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]
```

Создание документов актеров в коллекции persons



Добавлены “Bruce Willis”, “Keanu Reeves”, “Sandra Bullock”, “Quentin Tarantino”

db.persons.insertOne (

{

"id": 0000246,

"name": "Bruce Willis",

"headshot": "https://m.media-amazon.com/images/M/MV5BMjA0MjMzMTE5OF5BMl5BanBnXkFtZTcwMzQ2ODE3Mw@@._V1_UY98_CR8,0,67,98_AL_.jpg",

"birthDate": "1955-03-19",

"tradeMarks": ['Frequently plays a man who suffered a tragedy, had lost something or had a crisis of confidence or conscience.',

'Frequently plays likeable wisecracking heroes with a moral centre',

'Headlines action-adventures, often playing a policeman, hitman or someone in the military',

'Often plays men who get caught up in situations far beyond their control',

'Sardonic one-liners',

'Shaven head',

'Distinctive, gravelly voice',

'Smirky grin.',

'Known for playing cynical anti-heroes with unhappy personal lives'],

"actedInMovies": [

{ "movieId": "0110912", "title": "Pulp Fiction"},

{ "movieId": "1606378", "title": "A Good Day to Die Hard"},

{ "movieId": "0217869", "title": "Unbreakable"},

{ "movieId": "0377917", "title": "The Fifth Element"},

{ "movieId": "0112864", "title": "Die Hard: With a Vengeance"}]

})

db.persons.insertOne (

{

"id": 0000206,

"name": "Keanu Reeves",

"headshot": "https://m.media-amazon.com/images/M/MV5BMjA0MjMzMTE5OF5BMl5BanBnXkFtZTcwMzQ2ODE3Mw@@._V1_UY98_CR8,0,67,98_AL_.jpg",

"birthDate": "1955-03-19",

"tradeMarks": ['Intense contemplative gaze',

```

        'Deep husky voice',
        'Known for playing stoic reserved characters'],
    "actedInMovies": [
        { "movieId": "0133093", "title": "The Matrix" },
        { "movieId": "0234215", "title": "The Matrix Reloaded" },
        { "movieId": "0111257", "title": "Speed" }
    ]
})
db.persons.insertOne (
{
    "id": 0000113,
    "name": "Sandra Bullock",
    "headshot": "https://m.media-
amazon.com/images/M/MV5BMTI5NDY5NjU3NF5BMl5BanBnXkFtZTcwMzQ
0MTMyMw@@._V1_UX67_CR0,0,67,98_AL_.jpg",
    "birthDate": "1964-07-26",
    "actedInMovies": [
        { "movieId": "2737304", "title": "Bird Box" },
        { "movieId": "0120179", "title": "Speed 2: Cruise Control" },
        { "movieId": "0111257", "title": "Speed" },
        { "movieId": "0212346", "title": "Miss Congeniality" }
    ]
})
db.persons.insertOne (
{
    "id": 0000233,
    "name": "Quentin Tarantino",
    "headshot": "https://m.media-
amazon.com/images/M/MV5BMTgyMjI3ODAzNl5BMl5BanBnXkFtZTcwNzY2
MDYxOQ@@._V1_UX67_CR0,0,67,98_AL_.jpg",

```

"birthDate": "1963-03-27",

"tradeMarks": ['Lead characters usually drive General Motors vehicles, particularly Chevrolet and Cadillac, such as Jules 1974 Nova and Vincents 1960s Malibu.'],

'Briefcases and suitcases play an important role in Pulp Fiction (1994), Reservoir Dogs (1992), Jackie Brown (1997), True Romance (1993) and Kill Bill: Vol. 2 (2004).',

'Makes references to cult movies and television',

'Frequently works with Harvey Keitel, Tim Roth, Michael Madsen, Uma Thurman, Michael Bowen, Samuel L. Jackson, Michael Parks and Christoph Waltz.',

'His films usually have a shot from inside an automobile trunk',

'He always has a Dutch element in his films: The opening tune, Little Green Bag, in Reservoir Dogs (1992) was performed by George Baker Selection and written by Jan Gerbrand Visser and Benjamino Bouwens who are all Dutch. The character Freddy Newandyke, played by Tim Roth is a direct translation to a typical Dutch last name, Nieuwendijk. The code name of Tim Roth is Mr. Orange, the royal color of Holland and the last name of the royal family. The Amsterdam conversation in Pulp Fiction (1994), Vincent Vega smokes from a Dutch tobacco shag (Drum), the mentioning of Rutger Hauer in Jackie Brown (1997), the brides name is Beatrix, the name of the Royal Dutch Queen.',

'[The Mexican Standoff] All his movies (including True Romance (1993), which he only wrote and did not direct) feature a scene in which three or more characters are pointing guns at each other at the same time.',

'Often uses an unconventional storytelling device in his films, such as retrospect (Reservoir Dogs (1992)), non-linear (Pulp Fiction (1994)), or "chapter" format (Kill Bill: Vol. 1 (2003)).',

'His films will often include one long, unbroken take where a character is followed around somewhere.'],

"actedInMovies": [

{ "movieId": "0378194", "title": "Kill Bill: Vol. 2" },

{ "movieId": "0110912", "title": "Speed 2: Cruise Control" },

{ "movieId": "0116367", "title": "From Dusk Till Dawn" },

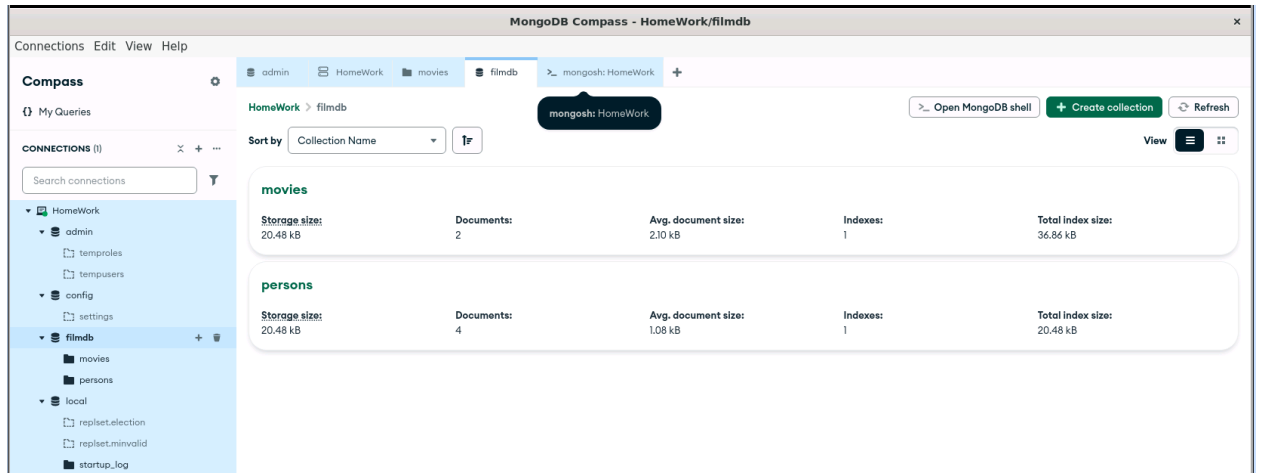
{ "movieId": "0119396", "title": "Jackie Brown" }

]

}}

Проверка

```
> db.persons.find().count()  
< 4
```



Запрос документов с помощью селектора запросов

Добавление новых документов (50 лучших фильмов из IMBD)

```
db.movies.insertMany([
    {"id": "0111161", "title": "The Shawshank Redemption", "genres":
["Drama"], "year": 1994, "rating": 9.2, "rank": 1},
    {"id": "0068646", "title": "The Godfather", "genres": ["Crime", "Drama"],
"year": 1972, "rating": 9.2, "rank": 2},
    {"id": "0071562", "title": "The Godfather: Part II", "genres": ["Crime",
"Drama"], "year": 1974, "rating": 9.0, "rank": 3},
    {"id": "0468569", "title": "The Dark Knight", "genres": ["Action", "Crime",
"Drama", "Thriller"], "year": 2008, "rating": 9.0, "rank": 4},
    {"id": "0050083", "title": "12 Angry Men", "genres": ["Drama"], "year":
1957, "rating": 8.9, "rank": 5},
    {"id": "0108052", "title": "Schindler's List", "genres": ["Biography",
"Drama", "History"], "year": 1993, "rating": 8.9, "rank": 6},
    {"id": "0167260", "title": "The Lord of the Rings: The Return of the King",
"genres": ["Adventure", "Drama", "Fantasy"], "year": 2003, "rating": 8.9, "rank":
7},
    {"id": "0060196", "title": "The Good, the Bad and the Ugly", "genres":
["Western"], "year": 1966, "rating": 8.8, "rank": 9},
    {"id": "0137523", "title": "Fight Club", "genres": ["Drama"], "year": 1999,
"rating": 8.8, "rank": 10},
    {"id": "4154796", "title": "Avengers: Endgame", "genres": ["Action",
"Adventure", "Fantasy", "Sci-Fi"], "year": 2019, "rating": 8.8, "rank": 11},
    {"id": "0120737", "title": "The Lord of the Rings: The Fellowship of the
Ring", "genres": ["Adventure", "Drama", "Fantasy"], "year": 2001, "rating": 8.8,
"rank": 12},
    {"id": "0109830", "title": "Forrest Gump", "genres": ["Drama", "Romance"],
"year": 1994, "rating": 8.7, "rank": 13},
    {"id": "0080684", "title": "Star Wars: Episode V - The Empire Strikes
Back", "genres": ["Action", "Adventure", "Fantasy", "Sci-Fi"], "year": 1980,
"rating": 8.7, "rank": 14},
    {"id": "1375666", "title": "Inception", "genres": ["Action", "Adventure",
"Sci-Fi", "Thriller"], "year": 2010, "rating": 8.7, "rank": 15},
```

{ "id": "0167261", "title": "The Lord of the Rings: The Two Towers",
"genres": ["Adventure", "Drama", "Fantasy"], "year": 2002, "rating": 8.7, "rank":
16},

{ "id": "0073486", "title": "One Flew Over the Cuckoo's Nest", "genres":
["Drama"], "year": 1975, "rating": 8.7, "rank": 17},

{ "id": "0099685", "title": "Goodfellas", "genres": ["Biography", "Crime",
"Drama"], "year": 1990, "rating": 8.7, "rank": 18},

{ "id": "0047478", "title": "Seven Samurai", "genres": ["Adventure",
"Drama"], "year": 1954, "rating": 8.6, "rank": 20},

{ "id": "0114369", "title": "Se7en", "genres": ["Crime", "Drama", "Mystery",
"Thriller"], "year": 1995, "rating": 8.6, "rank": 21 },

{ "id": "0317248", "title": "City of God", "genres": ["Crime", "Drama"],
"year": 2002, "rating": 8.6, "rank": 22},

{ "id": "0076759", "title": "Star Wars: Episode IV - A New Hope", "genres":
["Action", "Adventure", "Fantasy", "Sci-Fi"], "year": 1977, "rating": 8.6, "rank":
23},

{ "id": "0102926", "title": "The Silence of the Lambs", "genres": ["Crime",
"Drama", "Thriller"], "year": 1991, "rating": 8.6, "rank": 24},

{ "id": "0038650", "title": "It's a Wonderful Life", "genres": ["Drama",
"Family", "Fantasy"], "year": 1946, "rating": 8.6, "rank": 25},

{ "id": "0118799", "title": "Life Is Beautiful", "genres": ["Comedy",
"Drama", "Romance", "War"], "year": 1997, "rating": 8.6, "rank": 26},

{ "id": "0245429", "title": "Spirited Away", "genres": ["Animation",
"Adventure", "Family", "Fantasy", "Mystery"], "year": 2001, "rating": 8.5, "rank":
27},

{ "id": "0120815", "title": "Saving Private Ryan", "genres": ["Drama",
"War"], "year": 1998, "rating": 8.5, "rank": 28},

{ "id": "0114814", "title": "The Usual Suspects", "genres": ["Crime",
"Mystery", "Thriller"], "year": 1995, "rating": 8.5, "rank": 29},

{ "id": "0110413", "title": "L\u00e9on: The Professional", "genres":
["Action", "Crime", "Drama", "Thriller"], "year": 1994, "rating": 8.5, "rank": 30},

{ "id": "0120689", "title": "The Green Mile", "genres": ["Crime", "Drama",
"Fantasy", "Mystery"], "year": 1999, "rating": 8.5, "rank": 31 },

{ "id": "0816692", "title": "Interstellar", "genres": ["Adventure", "Drama",
"Sci-Fi"], "year": 2014, "rating": 8.5, "rank": 32},

{ "id": "0054215", "title": "Psycho", "genres": ["Horror", "Mystery", "Thriller"], "year": 1960, "rating": 8.5, "rank": 33 },

{ "id": "0120586", "title": "American History X", "genres": ["Drama"], "year": 1998, "rating": 8.5, "rank": 34 },

{ "id": "0021749", "title": "City Lights", "genres": ["Comedy", "Drama", "Romance"], "year": 1931, "rating": 8.5, "rank": 35 },

{ "id": "0034583", "title": "Casablanca", "genres": ["Drama", "Romance", "War"], "year": 1942, "rating": 8.5, "rank": 36 },

{ "id": "0064116", "title": "Once Upon a Time in the West", "genres": ["Western"], "year": 1968, "rating": 8.5, "rank": 37 },

{ "id": "0253474", "title": "The Pianist", "genres": ["Biography", "Drama", "Music", "War"], "year": 2002, "rating": 8.5, "rank": 38 },

{ "id": "0027977", "title": "Modern Times", "genres": ["Comedy", "Drama", "Family", "Romance"], "year": 1936, "rating": 8.5, "rank": 39 },

{ "id": "1675434", "title": "The Intouchables", "genres": ["Biography", "Comedy", "Drama"], "year": 2011, "rating": 8.5, "rank": 40 },

{ "id": "0407887", "title": "The Departed", "genres": ["Crime", "Drama", "Thriller"], "year": 2006, "rating": 8.5, "rank": 41 },

{ "id": "0088763", "title": "Back to the Future", "genres": ["Adventure", "Comedy", "Sci-Fi"], "year": 1985, "rating": 8.5, "rank": 42 },

{ "id": "0103064", "title": "Terminator 2: Judgment Day", "genres": ["Action", "Sci-Fi"], "year": 1991, "rating": 8.5, "rank": 43 },

{ "id": "2582802", "title": "Whiplash", "genres": ["Drama", "Music"], "year": 2014, "rating": 8.5, "rank": 44 },

{ "id": "0110357", "title": "The Lion King", "genres": ["Animation", "Adventure", "Drama", "Family", "Musical"], "year": 1994, "rating": 8.5, "rank": 45 },

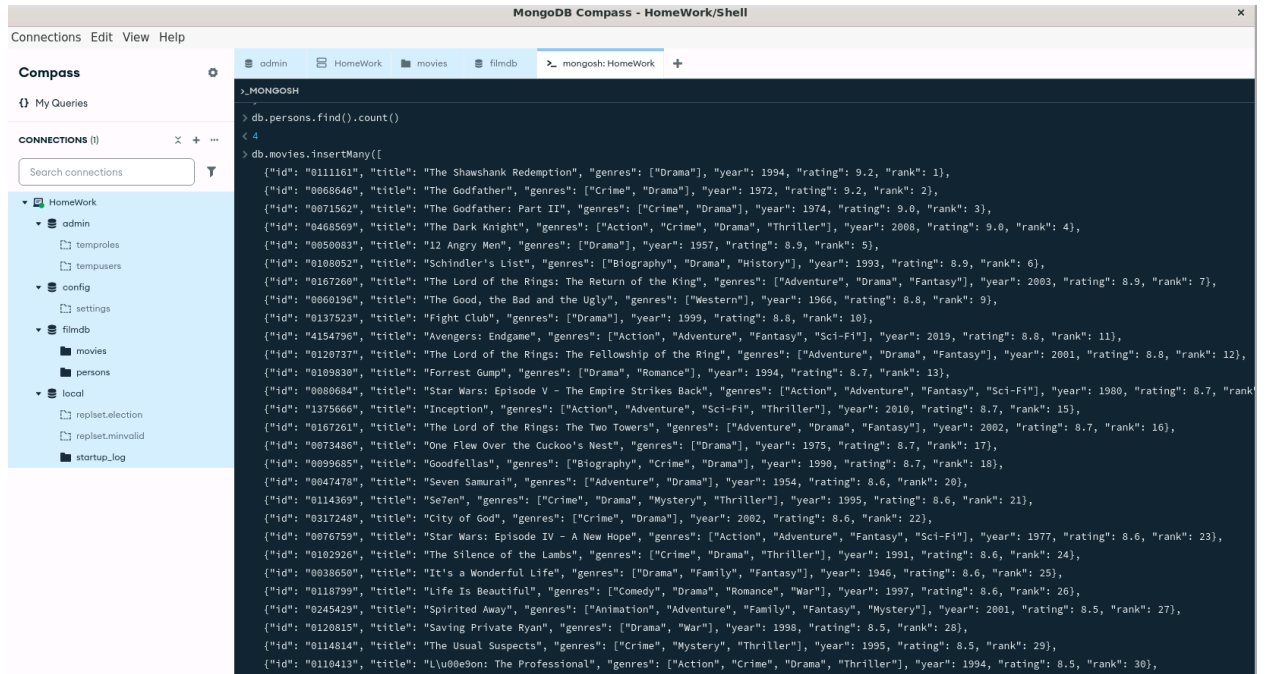
{ "id": "0047396", "title": "Rear Window", "genres": ["Mystery", "Thriller"], "year": 1954, "rating": 8.5, "rank": 46 },

{ "id": "0082971", "title": "Raiders of the Lost Ark", "genres": ["Action", "Adventure"], "year": 1981, "rating": 8.5, "rank": 47 },

{ "id": "0172495", "title": "Gladiator", "genres": ["Action", "Adventure", "Drama"], "year": 2000, "rating": 8.5, "rank": 48 },

```
{ "id": "0482571", "title": "The Prestige", "genres": ["Drama", "Mystery",  
"Sci-Fi", "Thriller"], "year": 2006, "rating": 8.5, "rank": 49},  
  
{ "id": "0078788", "title": "Apocalypse Now", "genres": ["Drama", "War"],  
"year": 1979, "rating": 8.4, "rank": 50}
```

)



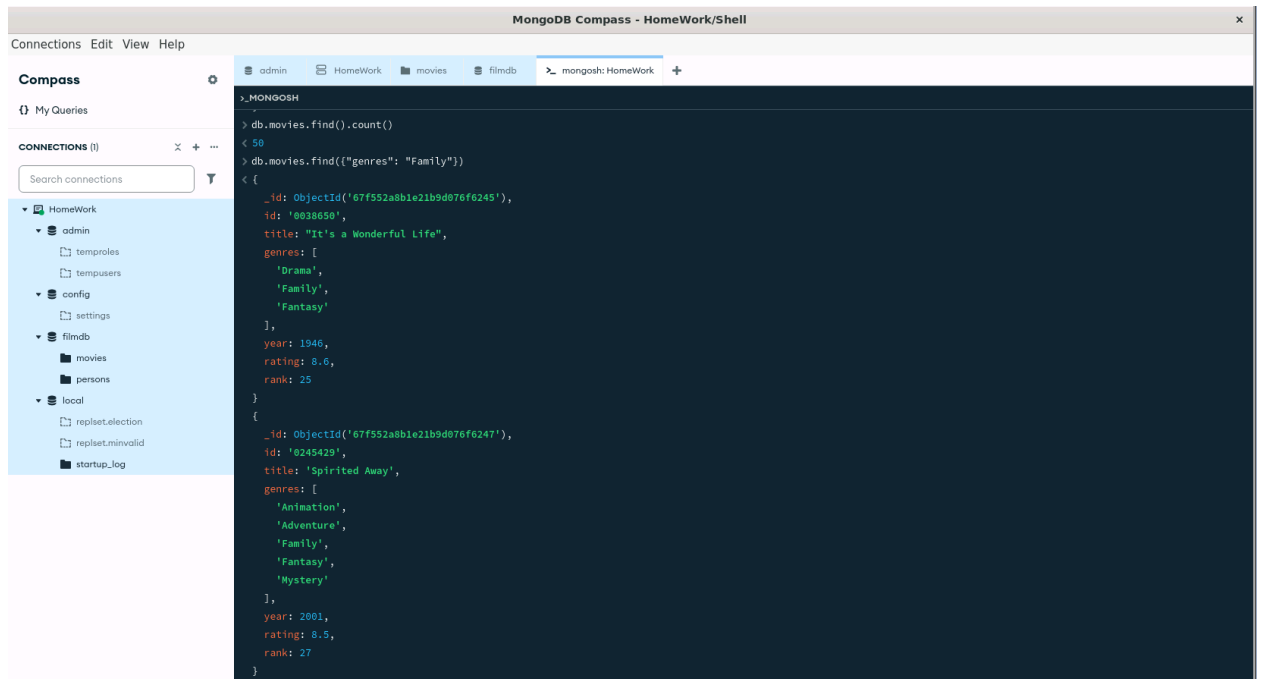
Проверка

db.movies.find().count()

```
> db.movies.find().count()  
< 50  
filddb > |
```

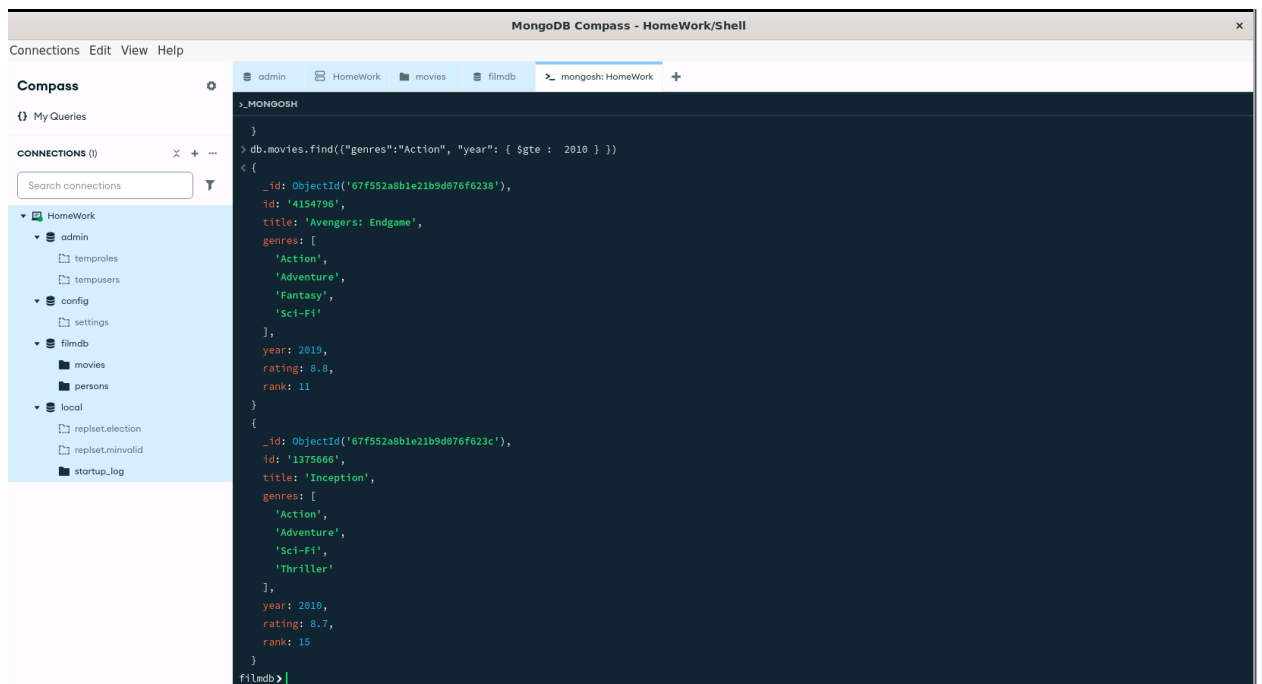
Все Family фильмы

`db.movies.find({"genres": "Family"})`



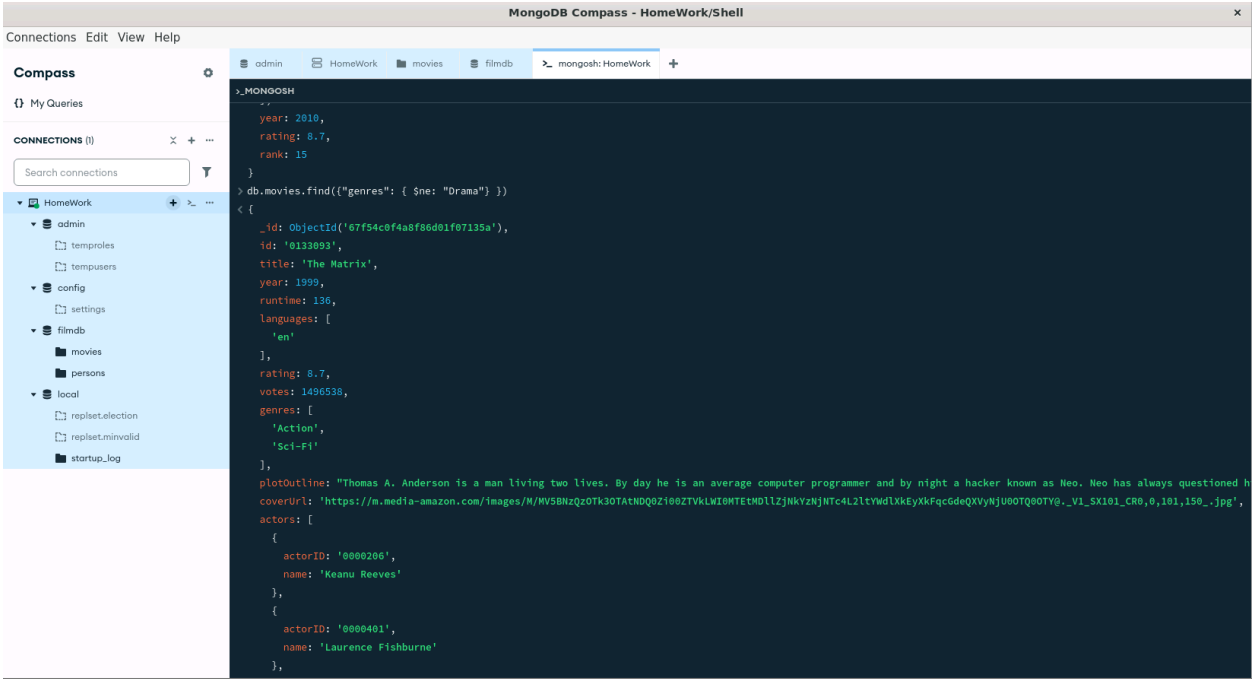
Все фильмы, которые были опубликованы в 2010 году и позже

`db.movies.find({"genres": "Action", "year": { $gte : 2010 } })`



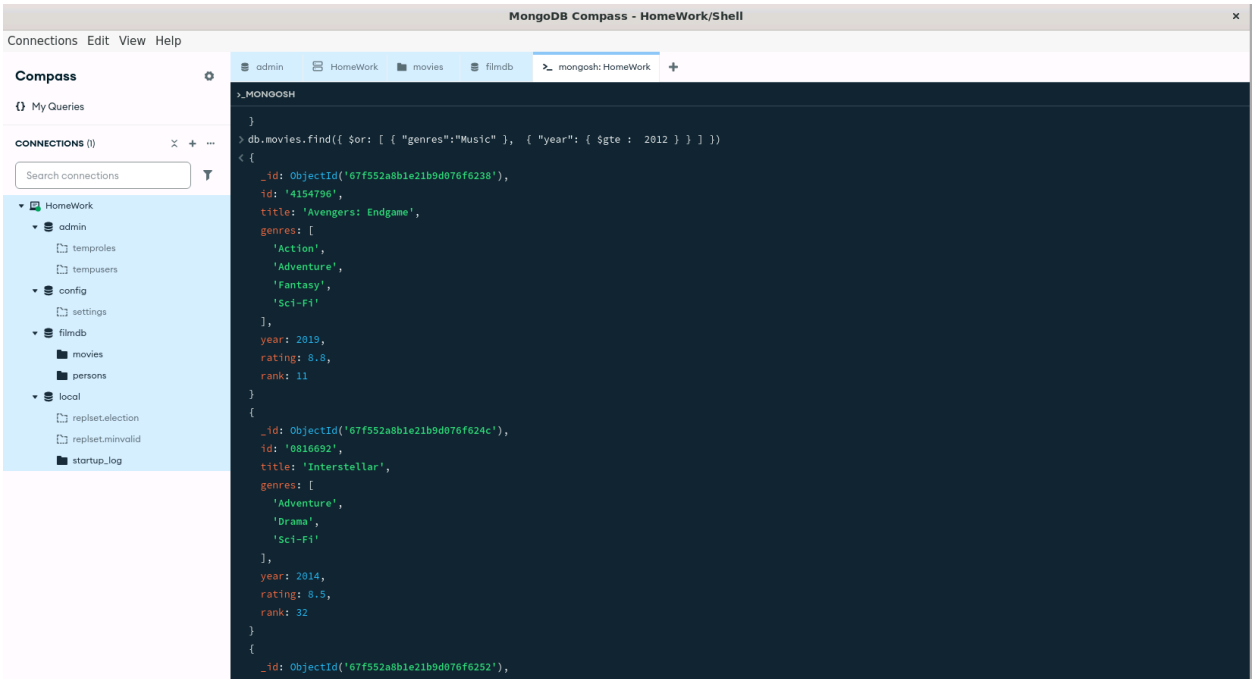
Все фильмы, которые не not относятся к жанру Drama

```
db.movies.find({"genres": { $ne: "Drama" } })
```



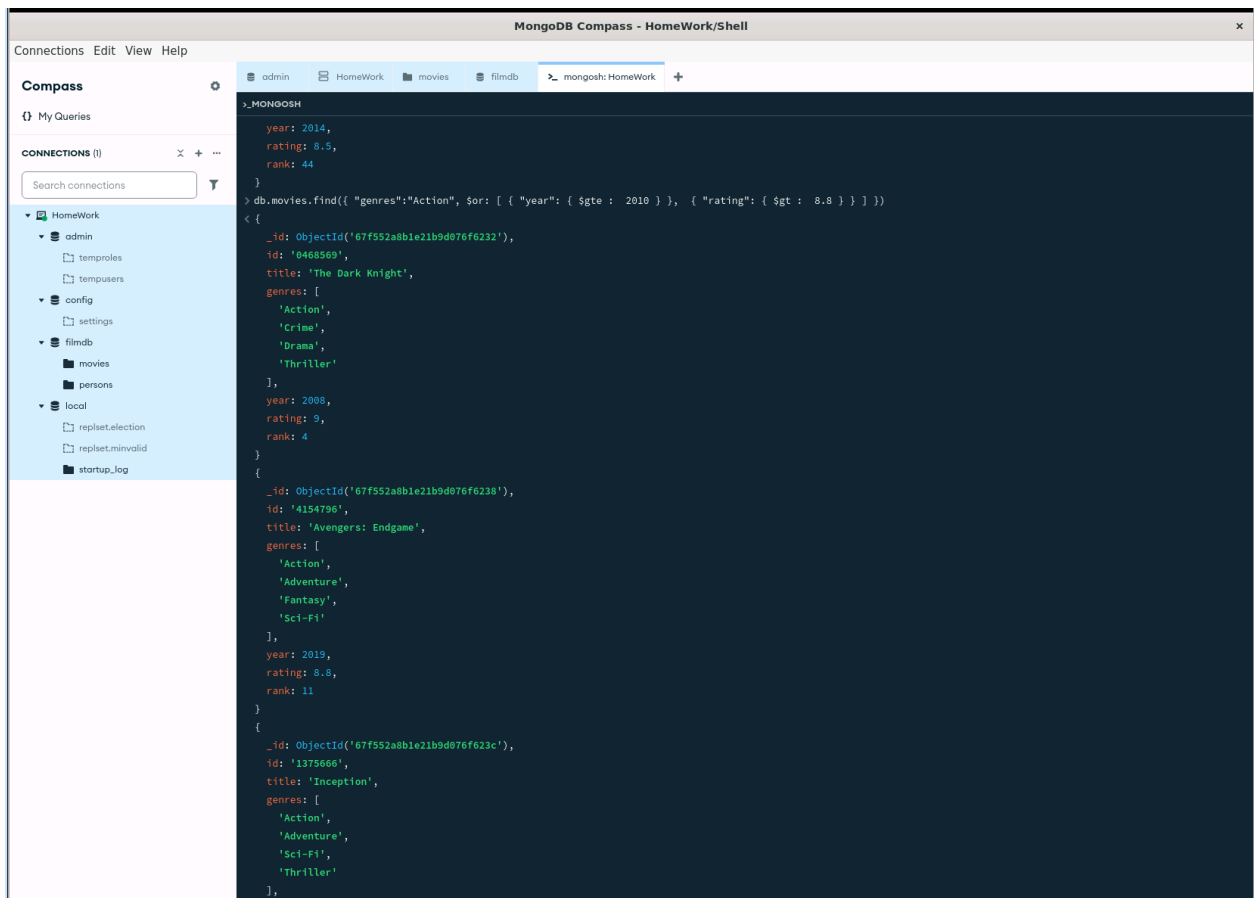
Все фильмы жанра Music OR, которые были выпущены в 2012 году или позже

```
db.movies.find({ $or: [ { "genres":"Music" }, { "year": { $gte : 2012 } } ] })
```



Все фильмы жанра Action AND которые были выпущены в 2010 году или позже OR имеют рейтинг выше 8.8

```
db.movies.find({ "genres":"Action", $or: [ { "year": { $gte : 2010 } }, { "rating": { $gt : 8.8 } } ] })
```



Индивидуальное задание. Вариант 13

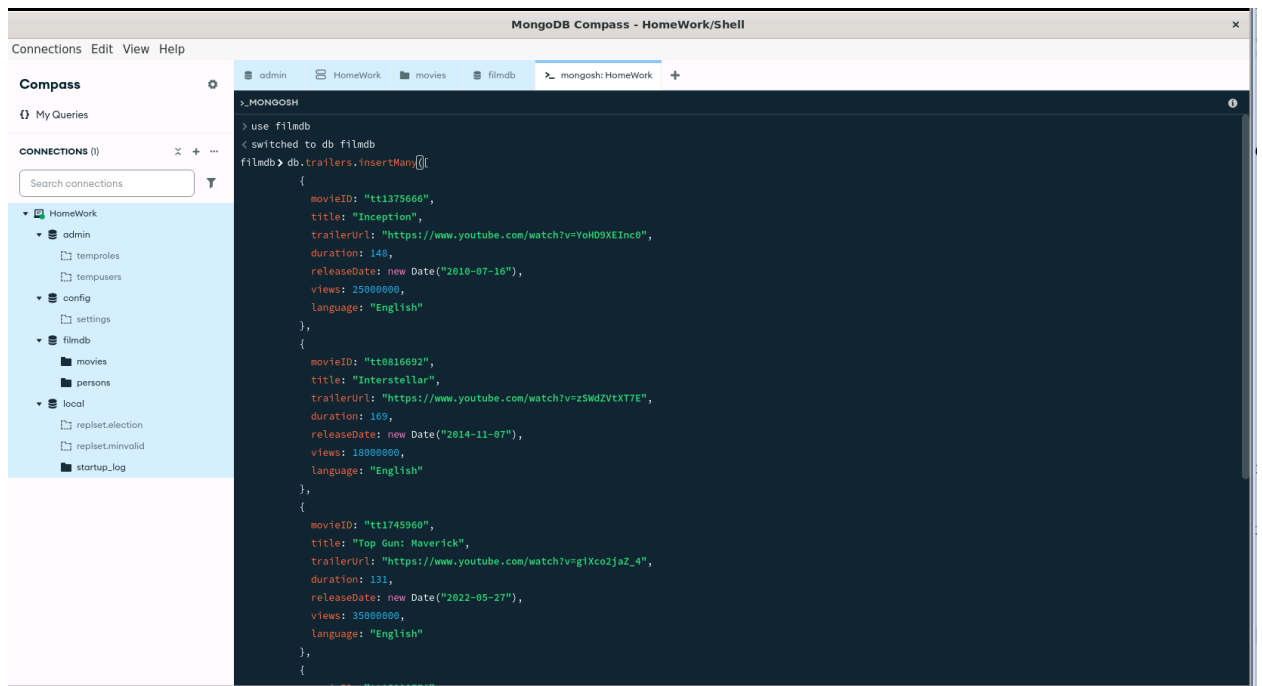
Задание:

1. Создайте коллекцию “trailers” с ссылками на трейлеры для новых фильмов

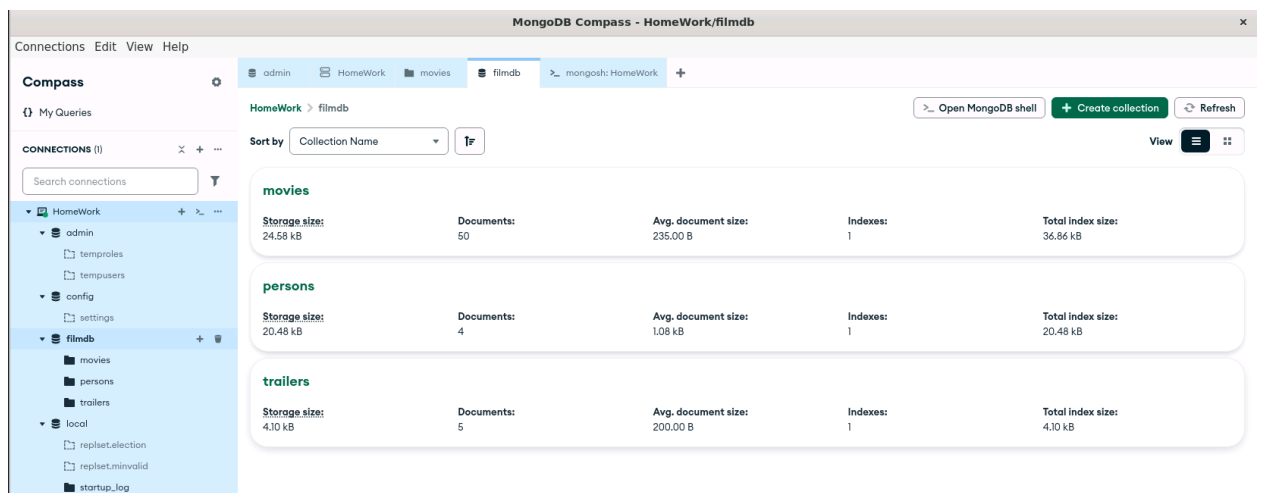
```
db.trailers.insertMany([
  {
    movieID: "tt1375666",
    title: "Inception",
    trailerUrl: "https://www.youtube.com/watch?v=YoHD9XElnc0",
    duration: 148,
    releaseDate: new Date("2010-07-16"),
    views: 25000000,
    language: "English"
  },
  {
    movieID: "tt0816692",
    title: "Interstellar",
    trailerUrl: "https://www.youtube.com/watch?v=zSWdZVtXT7E",
    duration: 169,
    releaseDate: new Date("2014-11-07"),
    views: 18000000,
    language: "English"
  },
  {
    movieID: "tt1745960",
    title: "Top Gun: Maverick",
    trailerUrl: "https://www.youtube.com/watch?v=giXco2jaZ_4",
    duration: 131,
    releaseDate: new Date("2022-05-27"),
```



```
views: 35000000,  
language: "English"  
},  
{  
  movieID: "tt15398776",  
  title: "Oppenheimer",  
  trailerUrl: "https://www.youtube.com/watch?v=uYPbbksJxIg",  
  duration: 180,  
  releaseDate: new Date("2023-07-21"),  
  views: 42000000,  
  language: "English"  
},  
{  
  movieID: "tt6718170",  
  title: "The Super Mario Bros. Movie",  
  trailerUrl: "https://www.youtube.com/watch?v=TnGl01FkMMo",  
  duration: 92,  
  releaseDate: new Date("2023-04-05"),  
  views: 38000000,  
  language: "English"  
}  
])
```

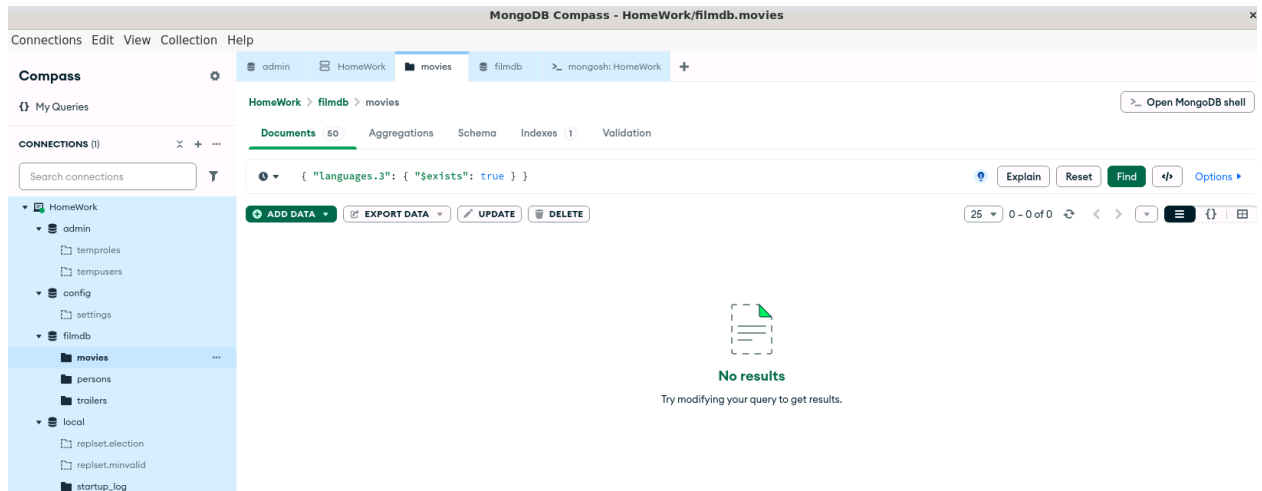


Проверка:



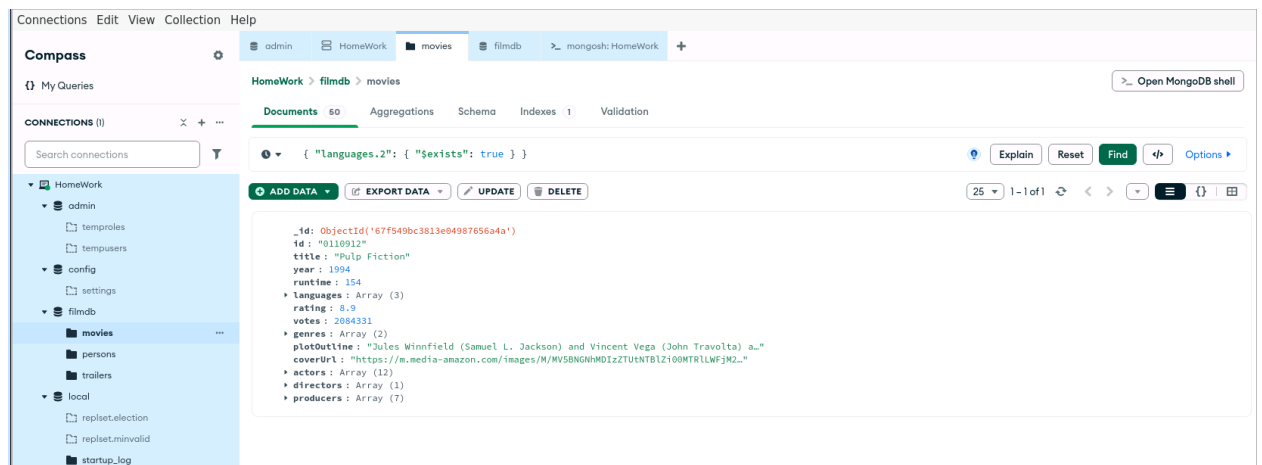
2. Найдите все фильмы с более чем 3 языками

```
{ "languages.3": { "$exists": true } }
```



Таких фильмов нет

Фильмы более чем с 2 языками



3. Добавьте поле “sequels” для фильмов, имеющих продолжения

Например, для фильма “The Matrix”

```
db.movies.updateOne(
```

```
  { title: "The Matrix" },
```

```
  {
```

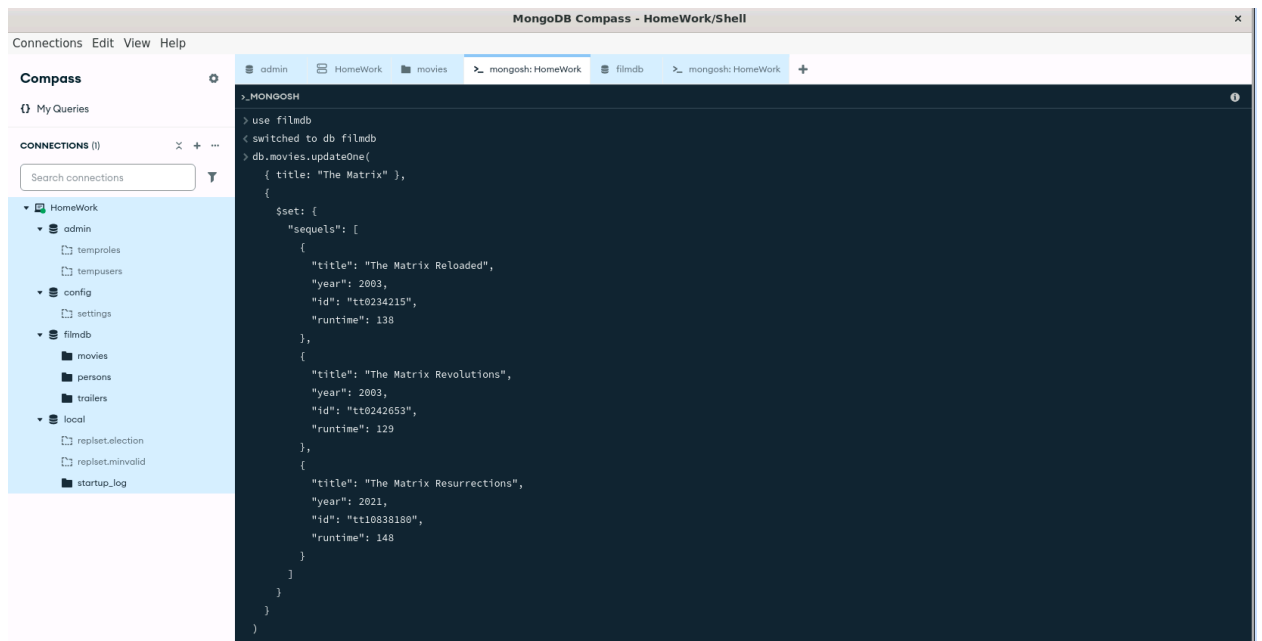
```
    $set: {
```

```
      "sequels": [
```

```
        {
```

```
          "title": "The Matrix Reloaded",
```

```
"year": 2003,  
"id": "tt0234215",  
"runtime": 138  
},  
{  
  "title": "The Matrix Revolutions",  
  "year": 2003,  
  "id": "tt0242653",  
  "runtime": 129  
},  
{  
  "title": "The Matrix Resurrections",  
  "year": 2021,  
  "id": "tt10838180",  
  "runtime": 148  
}  
]  
}  
}  
)
```

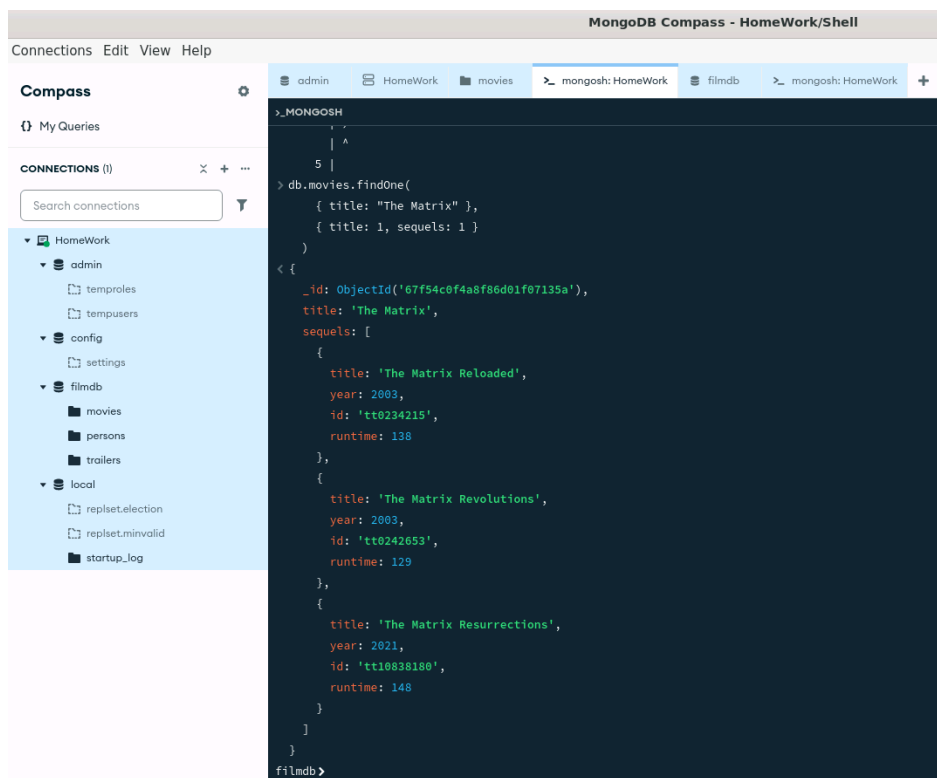


Проверка:

```

db.movies.findOne(
  { title: "The Matrix" },
  { title: 1, sequels: 1 }
)

```

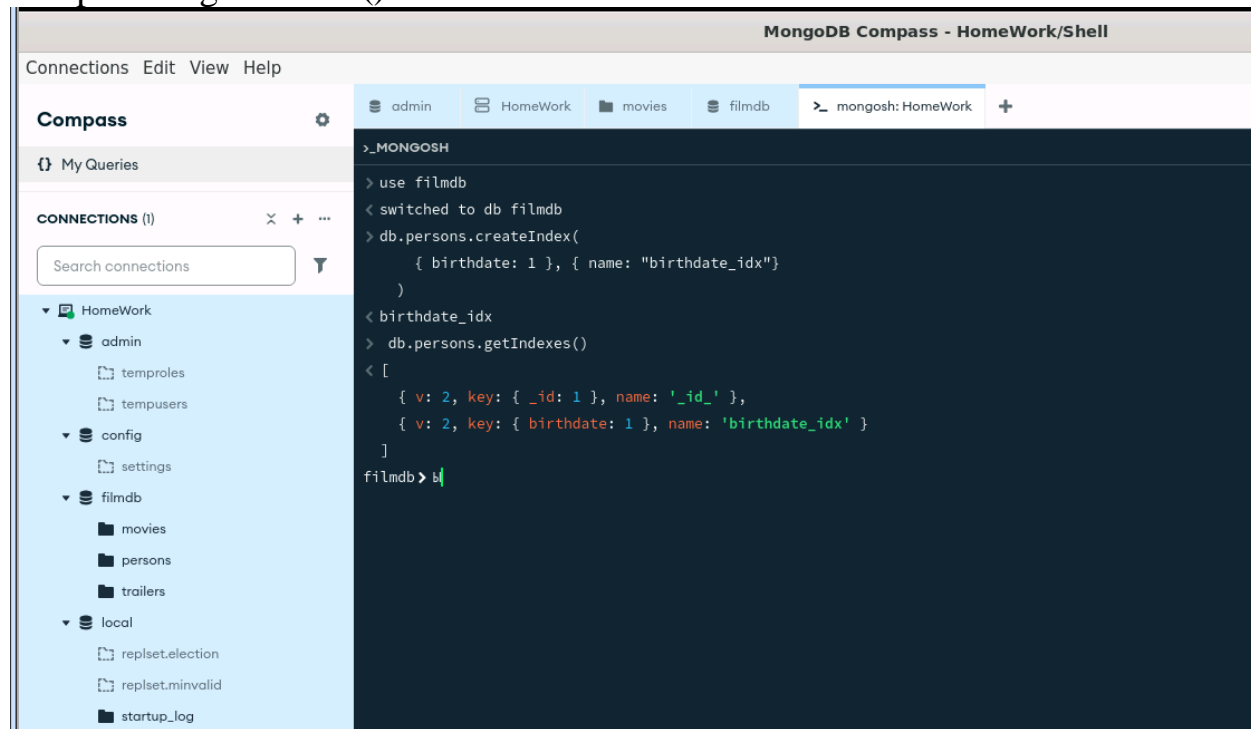


4. Создайте индекс по полю birthdate в коллекцию persons

```
db.persons.createIndex(  
  { birthdate: 1 }, { name: "birthdate_idx"  
})
```

Проверка

```
db.persons.getIndexes()
```



5. Рассчитайте процент фильмов каждого жанра от общего количества

```
db.movies.aggregate([  
  { $unwind: "$genres" }, // Разворачиваем массив жанров  
  {  
    $group: {  
      _id: "$genres", // Группируем по жанрам  
      count: { $sum: 1 }, // Считаем количество фильмов в каждом жанре  
      total: { $sum: 1 } // Считаем общее количество (пока по жанру)  
    }  
  },  
  {  
    $group: {
```

```

    _id: null, // Вторая группировка для расчета общего количества
    genres: { $push: { genre: "$_id", count: "$count" } },
    totalFilms: { $sum: "$count" } // Общее количество всех фильмов
  }
},
{ $unwind: "$genres" }, // Разворачиваем массив жанров
{
  $project: {
    _id: 0,
    genre: "$genres.genre",
    count: "$genres.count",
    percentage: {
      $multiply: [
        { $divide: ["$genres.count", "$totalFilms"] },
        100
      ]
    } // Рассчитываем процент
  }
},
{ $sort: { percentage: -1 } } // Сортируем по убыванию процента
)

```

MongoDB Compass - HomeWork/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

Search connections

HomeWork

- admin
 - temproles
 - tempusers
- config
 - settings
- filmdb
 - movies
 - persons
 - trailers
- local
 - replset.election
 - replset.minvalid
 - startup_log

```
>_MONGOSH
$project: {
  _id: 0,
  genre: "$genres.genre",
  count: "$genres.count",
  percentage: {
    $multiply: [
      { $divide: ["$genres.count", "$totalFilms"] },
      100
    ]
  } // Рассчитываем процент
},
{ $sort: { percentage: -1 } } // Сортируем по убыванию процента
])
< {
  genre: 'Drama',
  count: 36,
  percentage: 25.899280575539567
}
{
  genre: 'Adventure',
  count: 14,
  percentage: 10.071942446043165
}
{
  genre: 'Crime',
  count: 12,
  percentage: 8.633093525179856
}
{
  genre: 'Thriller',
```

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

Search connecti

HomeWork

- admin
 - temproles
 - tempusers
- config
 - settings
- filmdb
 - movies
 - persons
 - trailers
- local
 - replset.election
 - replset.minvalid
 - startup_log

```
>_MONGOSH
{ $sort: { percentage: -1 } } // Сортируем по убыванию процента
])
< {
  genre: 'Drama',
  count: 36,
  percentage: 25.899280575539567
}
{
  genre: 'Adventure',
  count: 14,
  percentage: 10.071942446043165
}
{
  genre: 'Crime',
  count: 12,
  percentage: 8.633093525179856
}
{
  genre: 'Thriller',
  count: 10,
  percentage: 7.194244604316546
}
{
  genre: 'Action',
  count: 10,
  percentage: 7.194244604316546
}
{
  genre: 'Sci-Fi',
  count: 9,
  percentage: 6.474820143884892
}
{
  genre: 'Fantasy',
  count: 9,
  percentage: 6.474820143884892
}
{
  genre: 'Mystery',
  count: 7,
  percentage: 5.0359712230215825
}
{
```


Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

×

+

⋮

Search connections

▼

HomeWork

▼

admin

temproles

tempusers

▼

config

settings

▼

filmdb

movies

persons

trailers

▼

local

replset.election

replset.minvalid

startup_log

admin

HomeWork

movies

filmdb

> mongosh: HomeWork

+

> MONGOSH

count: 5,

percentage: 6.474820143884892

}

{

genre: 'Mystery',

count: 7,

percentage: 5.0359712230215825

}

{

genre: 'Romance',

count: 5,

percentage: 3.597122302158273

}

{

genre: 'War',

count: 5,

percentage: 3.597122302158273

}

{

genre: 'Comedy',

count: 5,

percentage: 3.597122302158273

}

{

genre: 'Biography',

count: 4,

percentage: 2.877697841726619

}

{

genre: 'Family',

count: 4,

percentage: 2.877697841726619

}

{

genre: 'Animation',

count: 2,

percentage: 1.4388489208633095

}

{

genre: 'Western',

count: 2,

percentage: 1.4388489208633095

}

{

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1) X + ...

Search connecti

▼ HomeWork

▼ admin

temproles

tempusers

▼ config

settings

▼ filmdb

movies

persons

trailers

▼ local

replset.election

replset.minvalid

startup_log

admin HomeWork movies filmdb mongosh: HomeWork +

>_MONGOSH

count: 3,
percentage: 3.597122302158273
}
{
genre: 'Biography',
count: 4,
percentage: 2.877697841726619
}
{
genre: 'Family',
count: 4,
percentage: 2.877697841726619
}
{
genre: 'Animation',
count: 2,
percentage: 1.4388489208633095
}
{
genre: 'Western',
count: 2,
percentage: 1.4388489208633095
}
{
genre: 'Music',
count: 2,
percentage: 1.4388489208633095
}
{
genre: 'Musical',
count: 1,
percentage: 0.7194244604316548
}
{
genre: 'Horror',
count: 1,
percentage: 0.7194244604316548
}
{
genre: 'History',
count: 1,
percentage: 0.7194244604316548
}
}
filmdb>

Заключение

В результате выполнения данной работы были получены практические и теоретические навыки работы с MongoDB, включая создание и модификацию коллекций, добавление и обновление документов, построение индексов для оптимизации запросов, а также выполнение агрегационных операций для сложного анализа данных.