SPRINT #9
NON-RELATIONAL DATABASE: MongoDB

student:
Leo Kalugin

Date: 22/01/2025

SUMMARY

I will be working with a database containing collections related to a cinematic entertainment application. I will be running some queries requested by the client, who will determine if you can handle the analytical part of the project related to their database.

RESULT

In this folder on the GitHub repository, you will find:

/db_data: json-files with data
- **users.json**: contains information about users, including names, email addresses, and encrypted passwords.
- **theaters.json**: contains theater data, such as theater ID, location (address, and geographic coordinates).
- **sessions.json**: contains user sessions, including user IDs and JWT tokens for authentication.
- **movies.json**: contains movie information, such as plot, genres, duration, cast, reviews, release year, directors, rating, and awards.
- **comments.json**: contains user comments about movies, with information about the author of the comment, movie ID, comment text, and date.

Sprint_9.pdf: descriptions and results obtained for each exercise

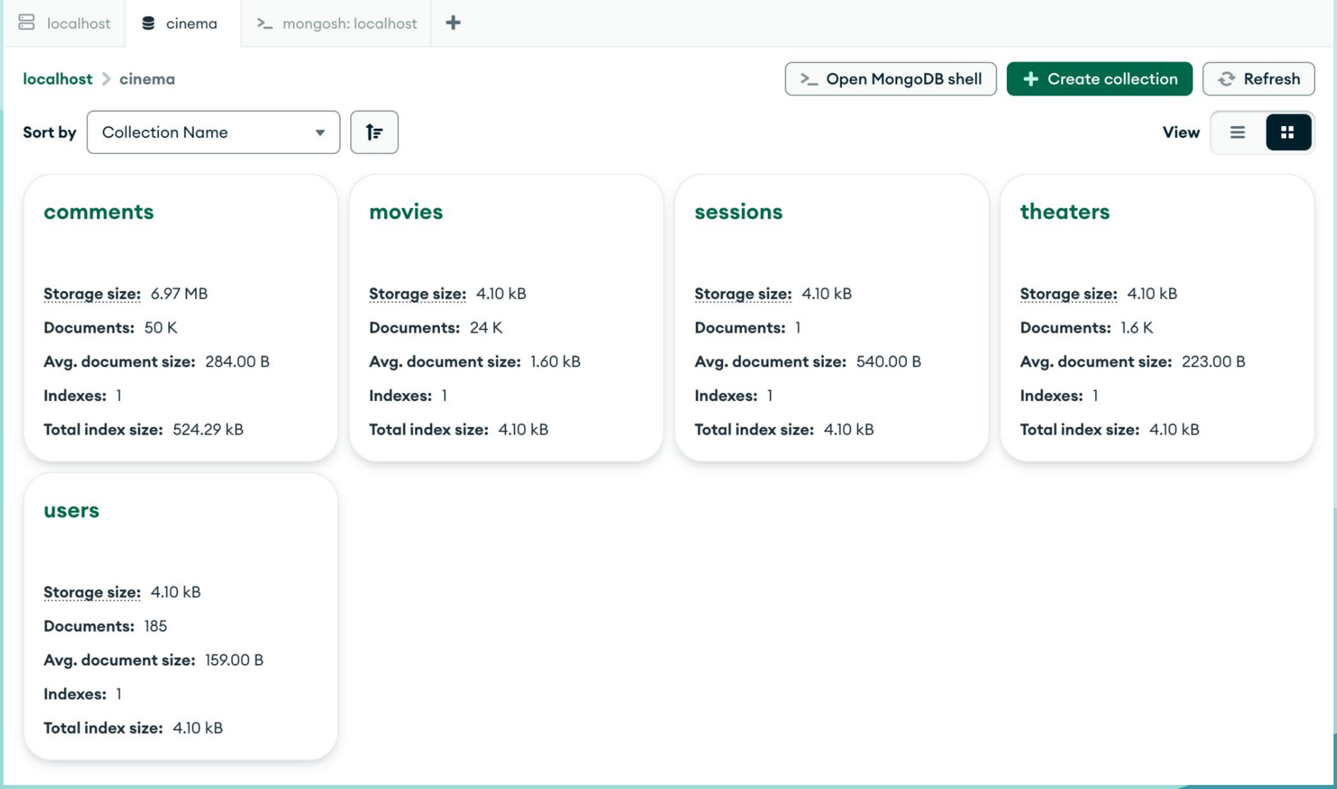
https://github.com/leocareer/DA_specialization/tree/main/Sprint_09

LEVEL 1 EXERCISE 0

Create a MongoDB database using attachments as collections.

SPRINT #9

22/01/2025



comments

Storage size: 6.97 MB
Documents: 50 K
Avg. document size: 284.00 B
Indexes: 1
Total index size: 524.29 kB

movies

Storage size: 4.10 kB
Documents: 24 K
Avg. document size: 1.60 kB
Indexes: 1
Total index size: 4.10 kB

sessions

Storage size: 4.10 kB
Documents: 1
Avg. document size: 540.00 B
Indexes: 1
Total index size: 4.10 kB

theaters

Storage size: 4.10 kB
Documents: 1.6 K
Avg. document size: 223.00 B
Indexes: 1
Total index size: 4.10 kB

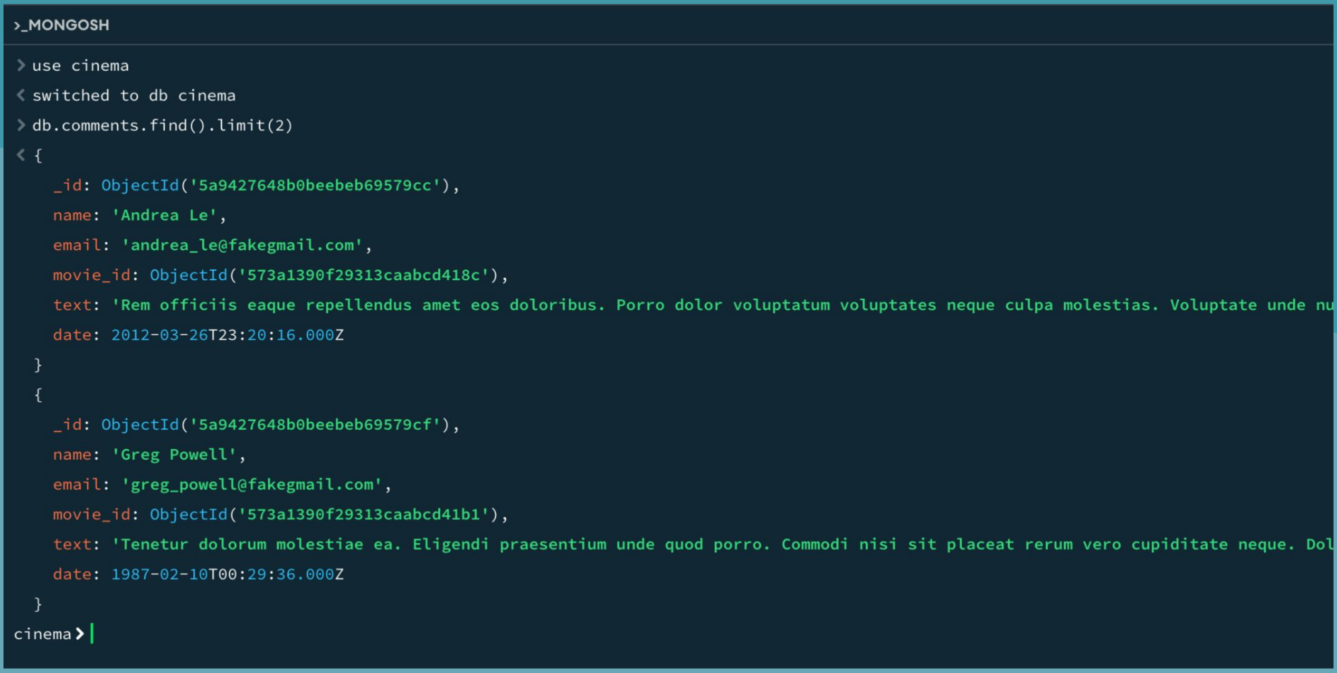
users

Storage size: 4.10 kB
Documents: 185
Avg. document size: 159.00 B
Indexes: 1
Total index size: 4.10 kB

LEVEL 1 EXERCISE 1.1

Show the first 2 comments in the database.

```
db.comments.find().limit(2)
```

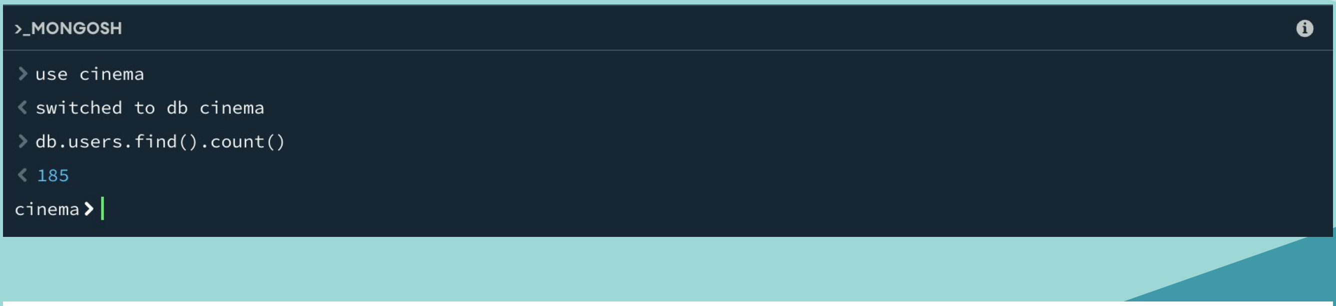


```
> use cinema
< switched to db cinema
> db.comments.find().limit(2)
< {
    _id: ObjectId('5a9427648b0beebeb69579cc'),
    name: 'Andrea Le',
    email: 'andrea_le@fakegmail.com',
    movie_id: ObjectId('573a1390f29313caabcd418c'),
    text: 'Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nu
    date: 2012-03-26T23:20:16.000Z
  }
  {
    _id: ObjectId('5a9427648b0beebeb69579cf'),
    name: 'Greg Powell',
    email: 'greg_powell@fakegmail.com',
    movie_id: ObjectId('573a1390f29313caabcd41b1'),
    text: 'Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Commodi nisi sit placeat rerum vero cupiditate neque. Dol
    date: 1987-02-10T00:29:36.000Z
  }
cinema >
```

LEVEL 1 EXERCISE 1.2

How many registered users do we have?

```
db.users.find().count()
```

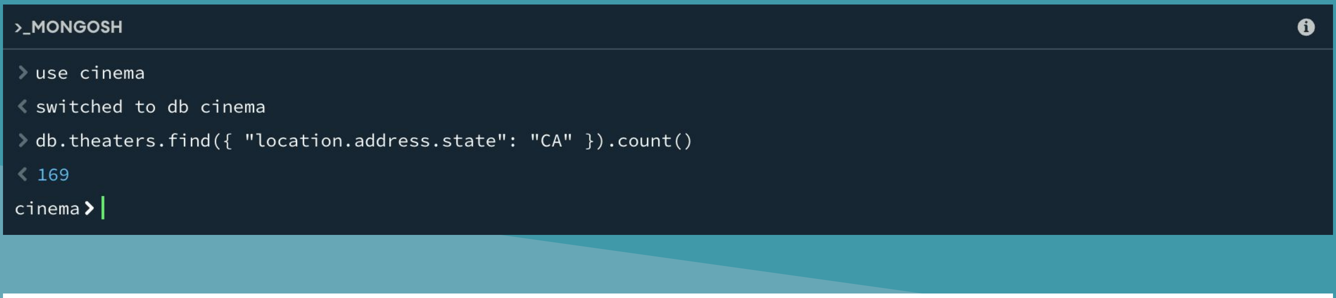


```
> use cinema
< switched to db cinema
> db.users.find().count()
< 185
cinema >
```

LEVEL 1 EXERCISE 1.3

How many movie theaters are there in California?

```
db.theaters.find({ "location.address.state": "CA" }).count()
```

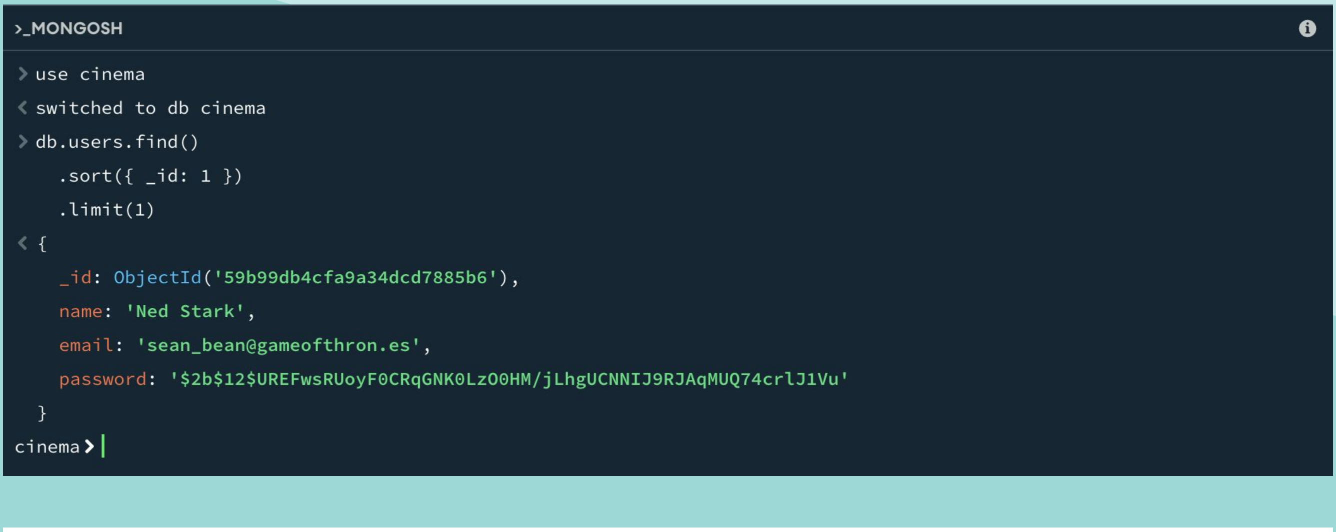


```
> use cinema
< switched to db cinema
> db.theaters.find({ "location.address.state": "CA" }).count()
< 169
cinema >
```

LEVEL 1 EXERCISE 1.4

Who was the first registered user?

```
db.users.find()
  .sort({ _id: 1 })
  .limit(1)
```

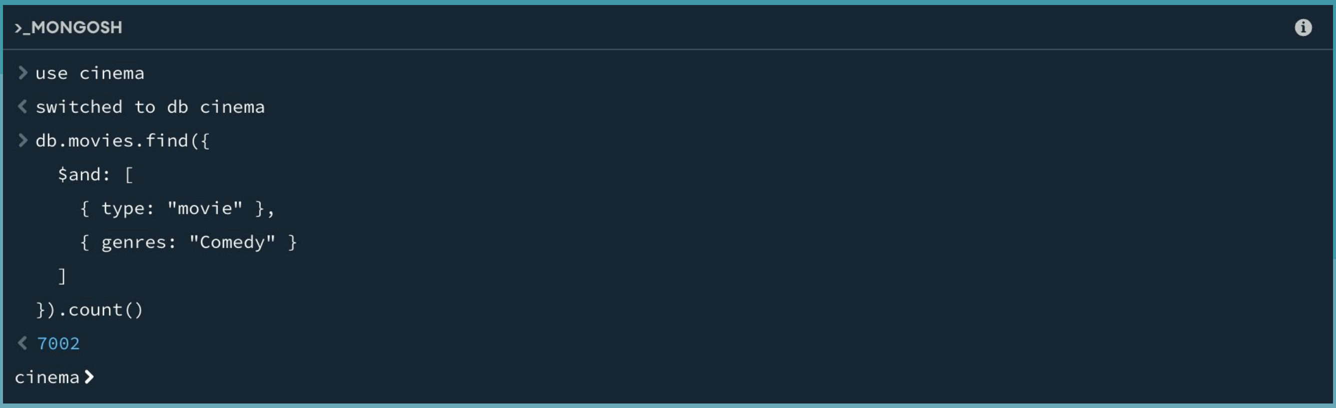


```
>_MONGOSH
> use cinema
< switched to db cinema
> db.users.find()
    .sort({ _id: 1 })
    .limit(1)
< {
    _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
    name: 'Ned Stark',
    email: 'sean_bean@gameofthron.es',
    password: '$2b$12$UREFwsRUoyF0CRqGNK0LzO0HM/jLhgUCNNIJ9RJAqMUQ74crlJ1Vu'
  }
cinema >
```

LEVEL 1 EXERCISE 1.5

How many comedy movies are in our database?

```
db.movies.find({
  $and: [
    { type: "movie" },
    { genres: "Comedy" }
  ]
}).count()
```

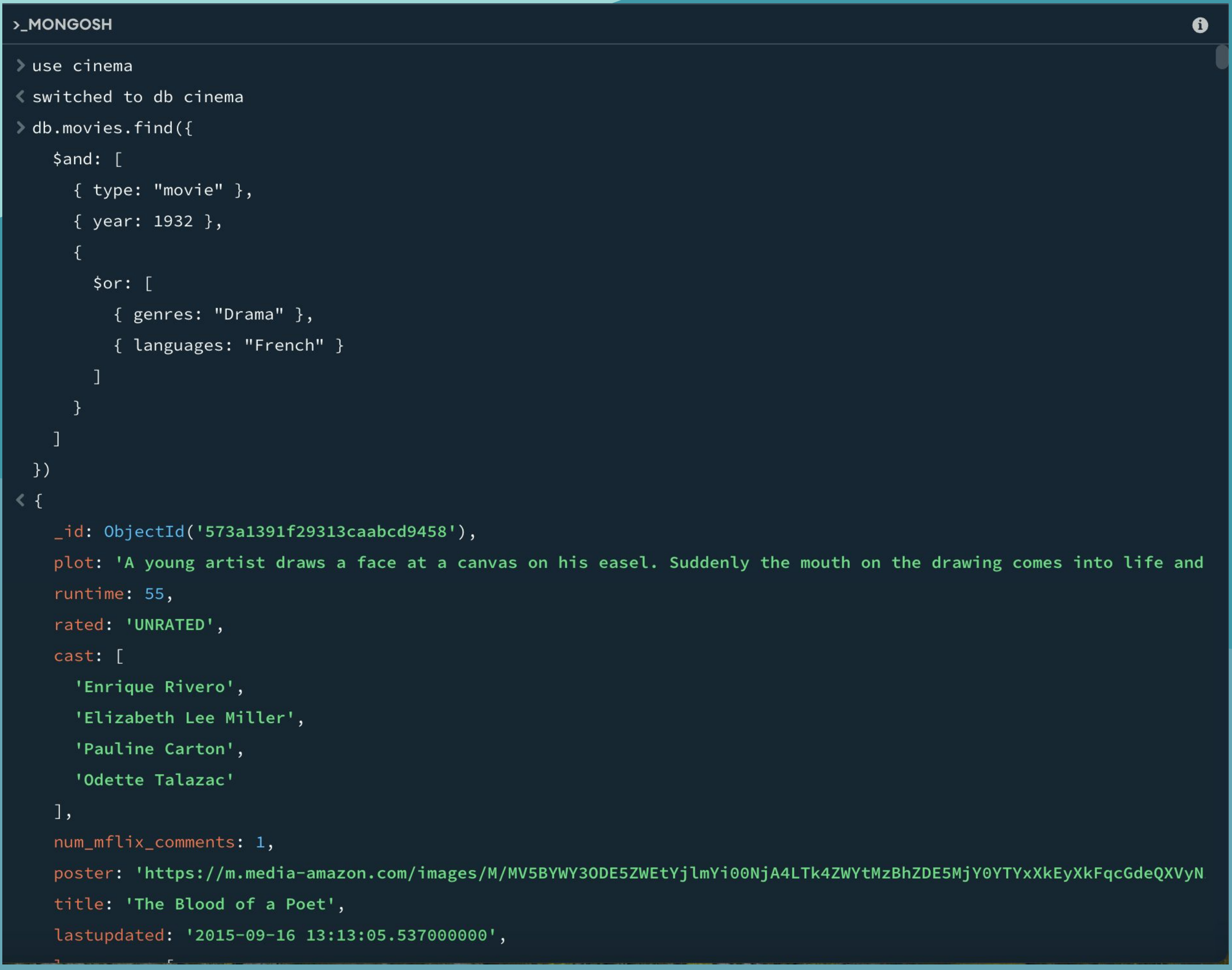


```
>_MONGOSH
> use cinema
< switched to db cinema
> db.movies.find({
    $and: [
      { type: "movie" },
      { genres: "Comedy" }
    ]
  }).count()
< 7002
cinema >
```

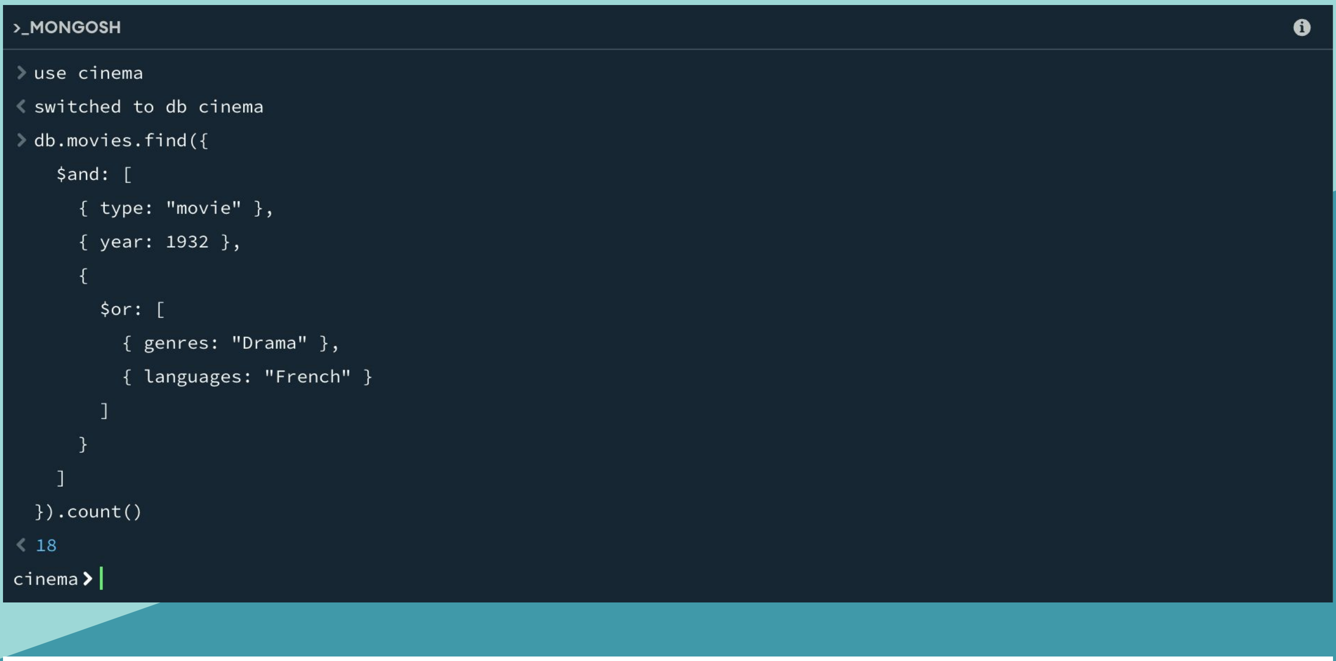
LEVEL 1 EXERCISE 2

Show all documents of films released in 1932, but of the genre drama or in French.

```
db.movies.find({
  $and: [
    { type: "movie" },
    { year: 1932 },
    {
      $or: [
        { genres: "Drama" },
        { languages: "French" }
      ]
    }
  ]
})
```



```
> use cinema
< switched to db cinema
> db.movies.find({
    $and: [
      { type: "movie" },
      { year: 1932 },
      {
        $or: [
          { genres: "Drama" },
          { languages: "French" }
        ]
      }
    ]
  })
< {
    _id: ObjectId('573a1391f29313caabcd9458'),
    plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and
    runtime: 55,
    rated: 'UNRATED',
    cast: [
      'Enrique Rivero',
      'Elizabeth Lee Miller',
      'Pauline Carton',
      'Odette Talazac'
    ],
    num_mflix_comments: 1,
    poster: 'https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTYxXkEyXkFqcGdeQXVyN
    title: 'The Blood of a Poet',
    lastupdated: '2015-09-16 13:13:05.537000000',
  db.movies.find({
    $and: [
      { type: "movie" },
      { year: 1932 },
      {
        $or: [
          { genres: "Drama" },
          { languages: "French" }
        ]
      }
    ]
  }).count()
```

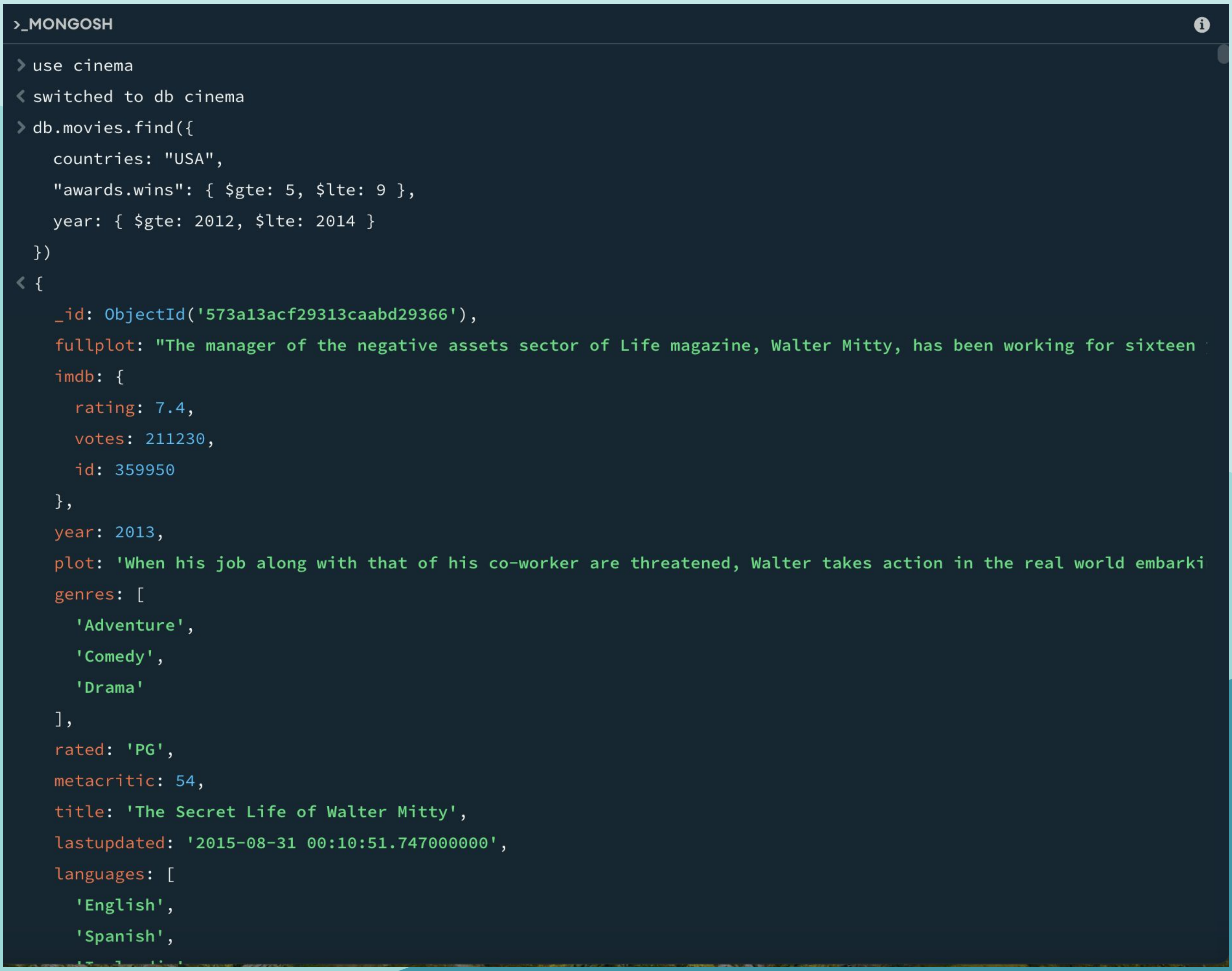


```
> use cinema
< switched to db cinema
> db.movies.find({
    $and: [
      { type: "movie" },
      { year: 1932 },
      {
        $or: [
          { genres: "Drama" },
          { languages: "French" }
        ]
      }
    ]
  }).count()
< 18
cinema >
```

LEVEL 1 EXERCISE 3

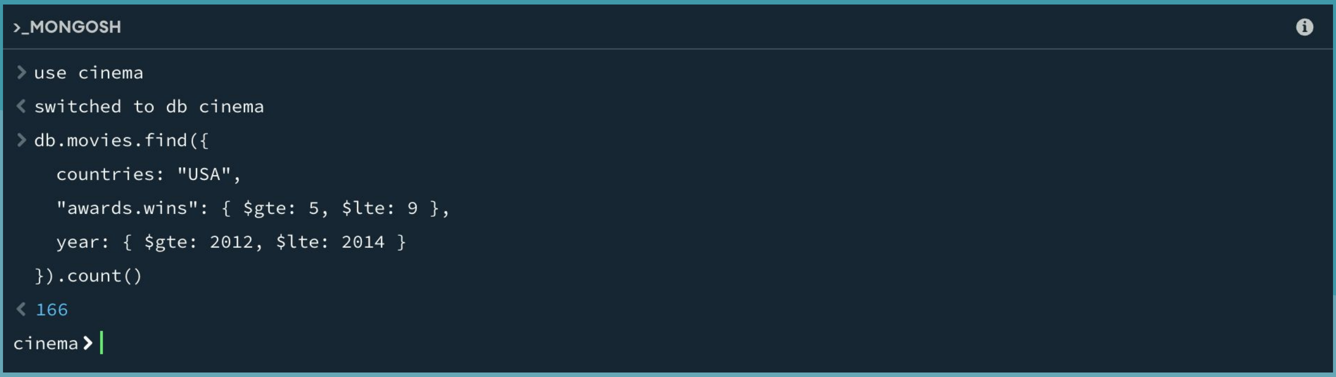
Show me all the records of American films that won 5-9 awards, made between 2012 and 2014.

```
db.movies.find({
  countries: "USA",
  "awards.wins": { $gte: 5, $lte: 9 },
  year: { $gte: 2012, $lte: 2014 }
})
```



```
> use cinema
< switched to db cinema
> db.movies.find({
    countries: "USA",
    "awards.wins": { $gte: 5, $lte: 9 },
    year: { $gte: 2012, $lte: 2014 }
  })
< {
    _id: ObjectId('573a13acf29313caabd29366'),
    fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen
    imdb: {
      rating: 7.4,
      votes: 211230,
      id: 359950
    },
    year: 2013,
    plot: 'When his job along with that of his co-worker are threatened, Walter takes action in the real world embarki
    genres: [
      'Adventure',
      'Comedy',
      'Drama'
    ],
    rated: 'PG',
    metacritic: 54,
    title: 'The Secret Life of Walter Mitty',
    lastupdated: '2015-08-31 00:10:51.747000000',
    languages: [
      'English',
      'Spanish',
```

```
db.movies.find({
  countries: "USA",
  "awards.wins": { $gte: 5, $lte: 9 },
  year: { $gte: 2012, $lte: 2014 }
}).count()
```

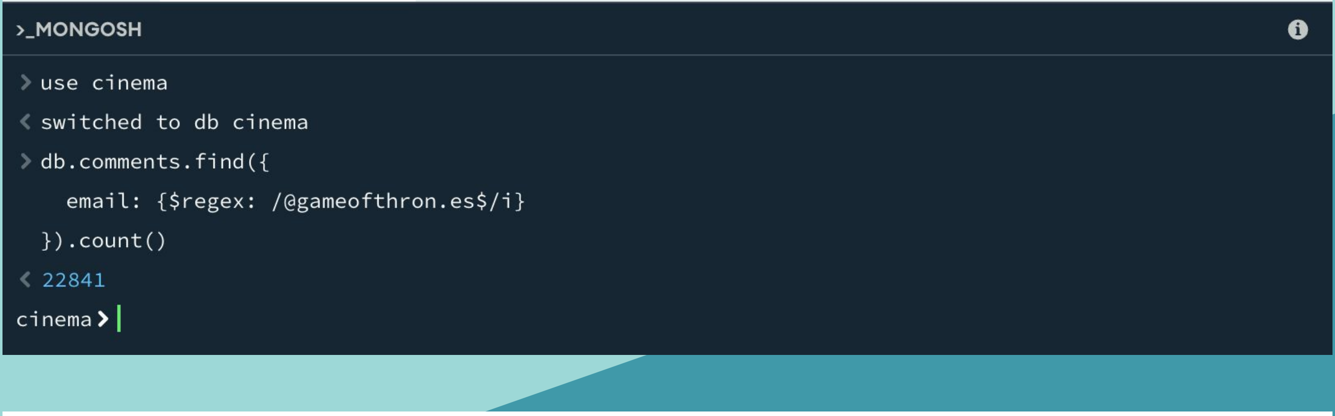


```
> use cinema
< switched to db cinema
> db.movies.find({
    countries: "USA",
    "awards.wins": { $gte: 5, $lte: 9 },
    year: { $gte: 2012, $lte: 2014 }
  }).count()
< 166
cinema >
```

LEVEL 2 EXERCISE 1

Count how many comments are written by a user using "GAMEOFTHRON.ES" as an email domain.

```
db.comments.find({
       email: {$regex: /@gameofthron.es$/i}
}).count()
```

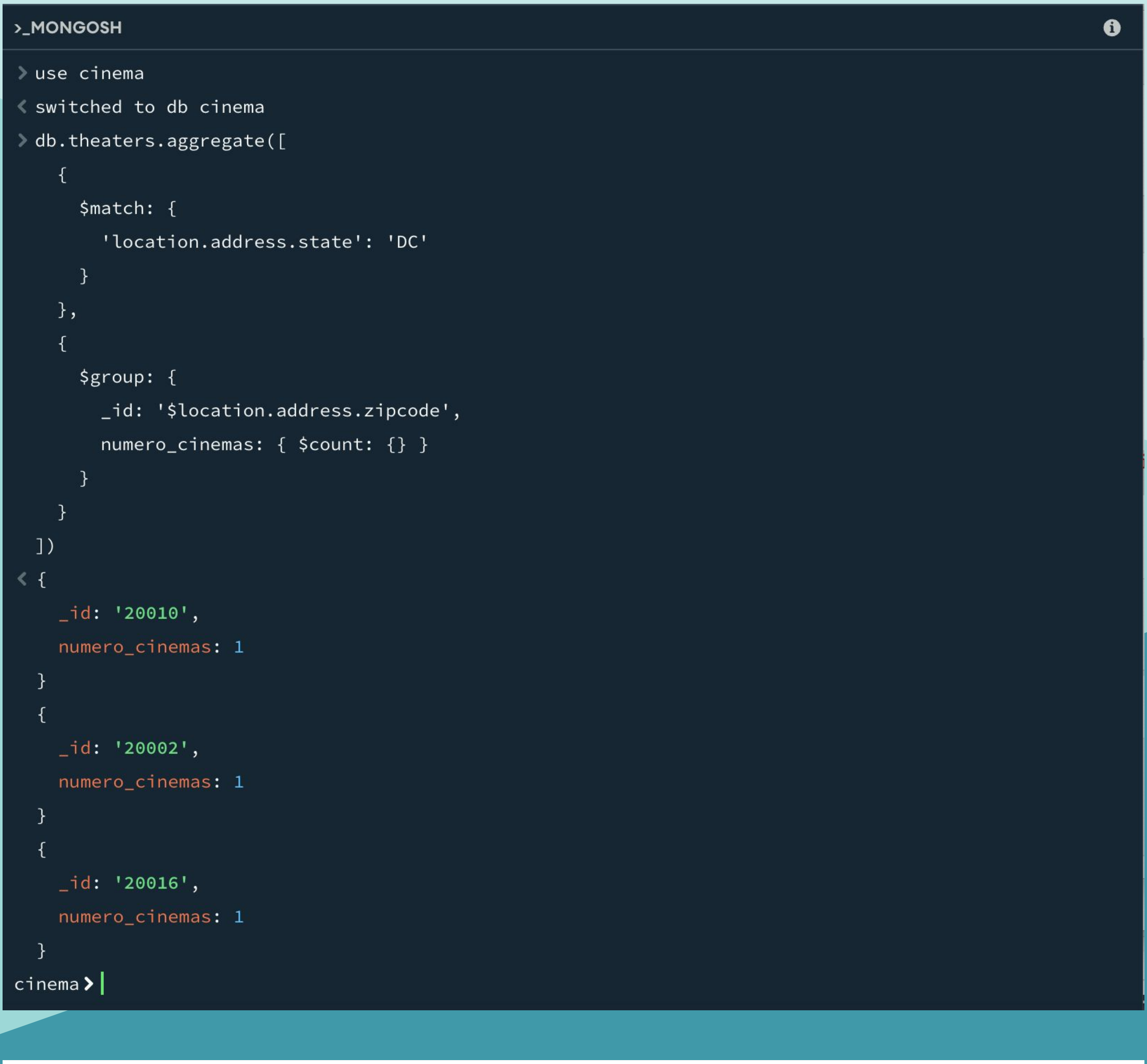


```
> use cinema
< switched to db cinema
> db.comments.find({
    email: {$regex: /@gameofthron.es$/i}
  }).count()
< 22841
cinema >
```

LEVEL 2 EXERCISE 2

How many movie theaters are there in each zip code located within the state of Washington D. C. (DC)?

```
db.theaters.aggregate([
  {
    $match: {
      'location.address.state': 'DC'
    }
  },
  {
    $group: {
      _id: '$location.address.zipcode',
      numero_cinemas: { $count: {} }
    }
  }
])
```

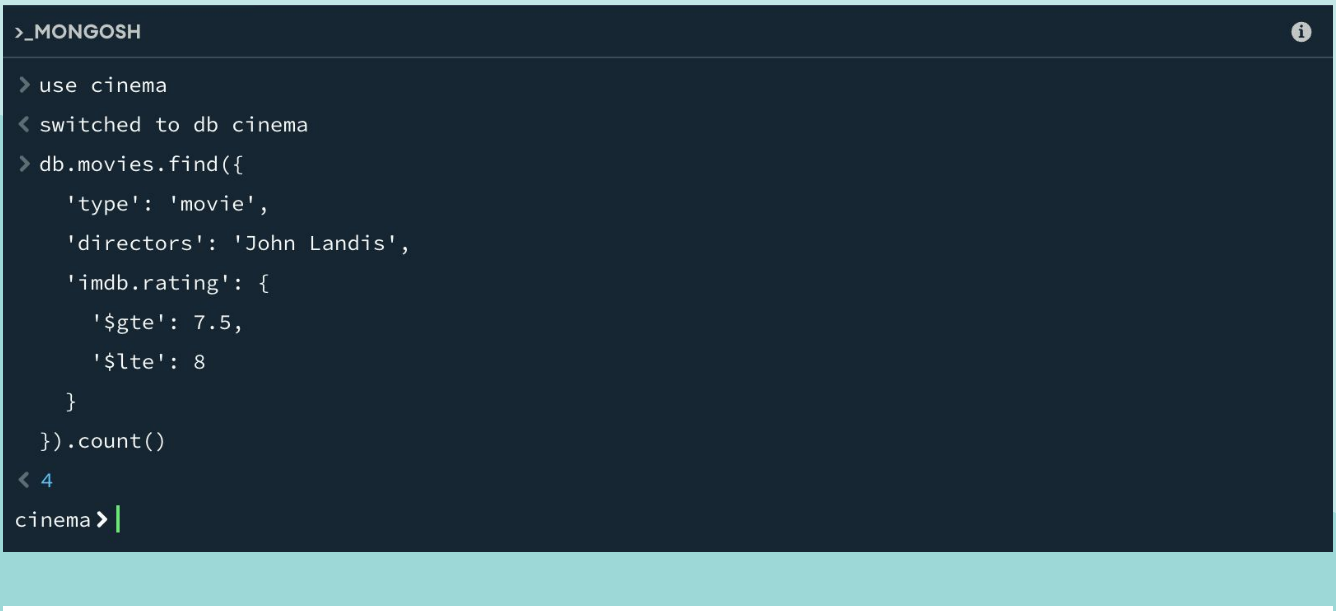


```
> use cinema
< switched to db cinema
> db.theaters.aggregate([
    {
      $match: {
        'location.address.state': 'DC'
      }
    },
    {
      $group: {
        _id: '$location.address.zipcode',
        numero_cinemas: { $count: {} }
      }
    }
  ])
< {
    _id: '20010',
    numero_cinemas: 1
  }
  {
    _id: '20002',
    numero_cinemas: 1
  }
  {
    _id: '20016',
    numero_cinemas: 1
  }
cinema >
```

LEVEL 3 EXERCISE 1

Find all films directed by John Landis with an IMDb (Internet Movie Database) rating between 7.5 and 8.

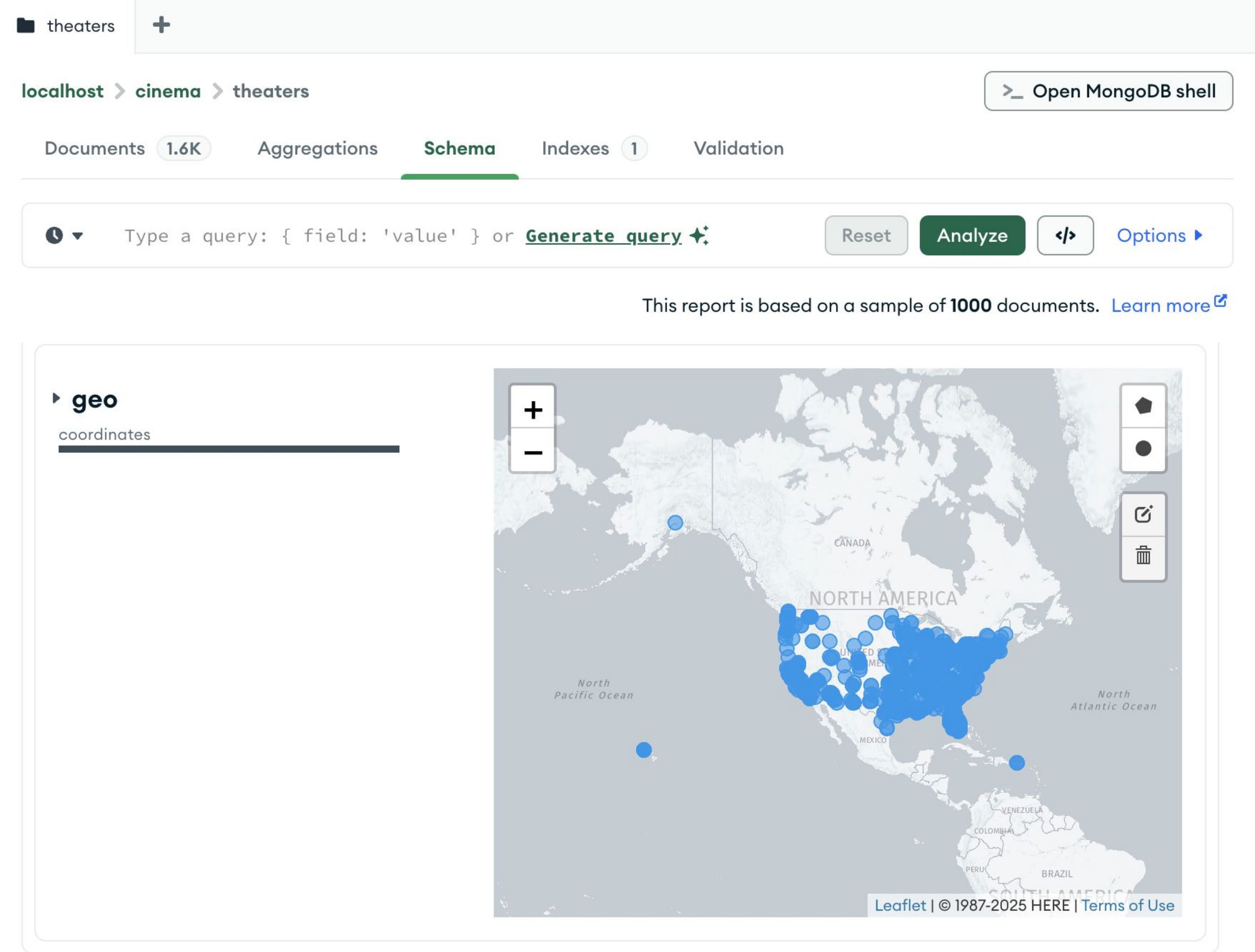
```
db.movies.find({
  'type': 'movie',
  'directors': 'John Landis',
  'imdb.rating': {
    '$gte': 7.5,
    '$lte': 8
  }
}).count()
```



```
> use cinema
< switched to db cinema
> db.movies.find({
    'type': 'movie',
    'directors': 'John Landis',
    'imdb.rating': {
      '$gte': 7.5,
      '$lte': 8
    }
  }).count()
< 4
cinema >
```

LEVEL 3 EXERCISE 1

Show the location of all the theaters in the database on a map.



This report is based on a sample of **1000** documents.