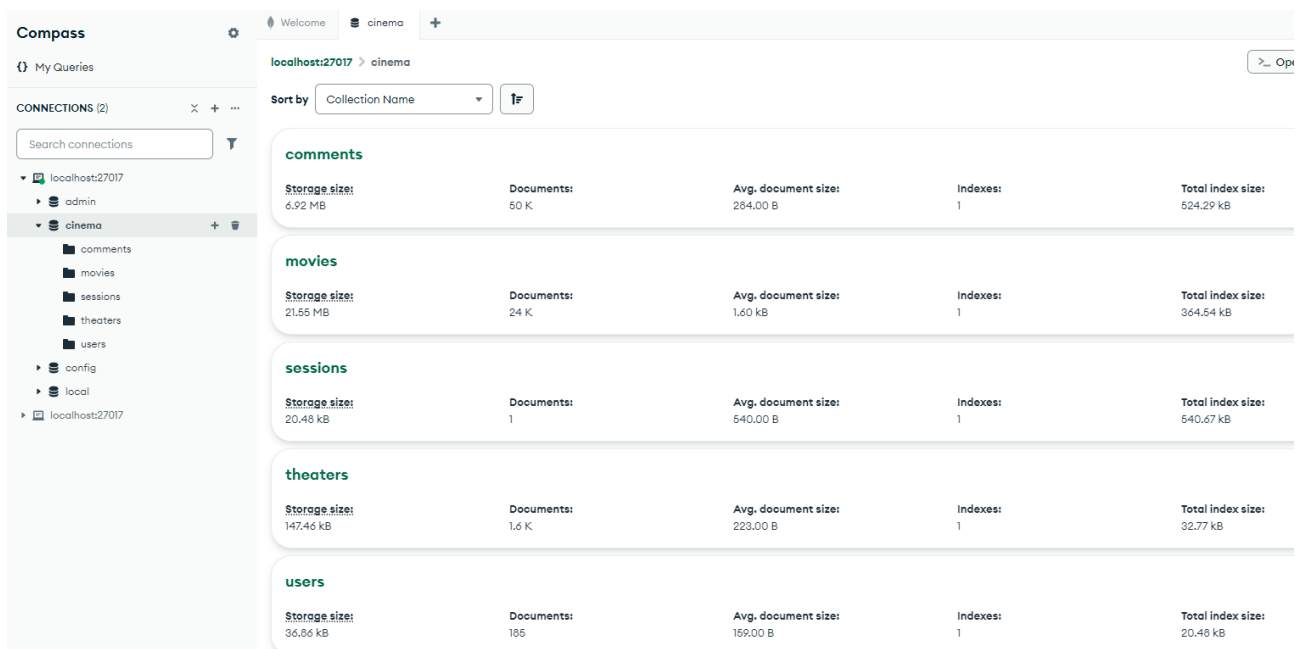


## Task 9.

### MongoDB

A database `'cinema'` containing information related to a movie entertainment application was created in MongoDB Compass and populated with data from the provided `.csv` files:

- `users`: user information, including names, emails, and encrypted passwords
- `theatres`: theater data, such as ID, location (address and geographic coordinates)
- `sessions`: user sessions, including user ID and JWT tokens for authentication
- `movies`: movie details, such as plot, genres, duration, cast, reviews, release year, directors, ratings, and awards
- `comments`: user comments about movies, with information about the comment author, movie ID, comment text, and date.



Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
<b>comments</b>	6.92 MB	50 K	284.00 B	1	524.29 kB
<b>movies</b>	21.55 MB	24 K	1.60 kB	1	364.54 kB
<b>sessions</b>	20.48 kB	1	540.00 B	1	540.67 kB
<b>theaters</b>	147.46 kB	1.6 K	223.00 B	1	32.77 kB
<b>users</b>	36.86 kB	105	159.00 B	1	20.48 kB

## Level 1.

### Exercise 1.1.

The first 2 comments in the data base.

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

```
> db.comments.find().limit(2)
< {
  _id: ObjectId('5a9427648b0beebeb69579cc'),
  name: 'Andrea Le',
  email: 'andrea_le@fakegmail.com',
  movie_id: ObjectId('573a1390f29313caabcd418c'),
  text: 'Rem officijs eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.',
  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. Dolorum nihil vero animi.',
  date: 1987-02-10T00:29:36.000Z
}
```

### Exercise 1.2.

How many users do we have registered?

```
db.users.countDocuments()
```

```
> _MONGOSH
> use cinema
< switched to db cinema
> db.users.countDocuments()
< 185
```

### Exercise 1.3.

How many movie theaters are there in the state of [California](#)?

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

```
> db.theaters.countDocuments({ "location.address.state": "CA" })
< 169
cinema>
```

### Exercise 1.4.

Who was the first user to register?

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

```
> db.users.find().sort({ _id: 1 }).limit(1)
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIj9RJAqMUQ74cr1J1Vu'
}
```

## Exercise 1.5.

How many **comedy movies** are in our database?

```
db.movies.countDocuments({ type: "movie", genres: "Comedy" })
```

```
> db.movies.countDocuments( {type: "movie", genres: "Comedy"} )  
< 7002
```

## Exercise 2.

All documents for **movies** produced in **1932**, but the genre is **drama** or they are in **French**.

*Let's show only title, year, genre and language of the movies.*

```
db.movies.find({  
  year: 1932,  
  type: "movie",  
  $or: [ { genres: "Drama" }, { languages: "French" } ]  
},  
{title: 1, year:1, genres:1, languages: 1}  
)
```

```
> db.movies.find({  
  year: 1932,  
  type: "movie",  
  $or: [ { genres: "Drama" }, { languages: "French" } ]  
},  
{title: 1, year:1, genres:1, languages: 1}  
)  
< {  
  _id: ObjectId('573a1391f29313caabcd9458'),  
  title: 'The Blood of a Poet',  
  languages: [  
    'French'  
  ],  
  year: 1932  
}  
{  
  _id: ObjectId('573a1392f29313caabcd99a3'),  
  genres: [  
    'Drama',  
    'Fantasy',  
    'Mystery'  
  ],  
  title: 'The Blue Light',  
  languages: [  
    'German',  
    'Italian'  
  ],  
  year: 1932  
}  
{  
  _id: ObjectId('573a1392f29313caabcd99e3'),  
  genres: [  
    'Drama'  
  ],  
  title: 'Broken Lullaby',  
  languages: [  
    'English'  
  ],  
  year: 1932  
}
```

```
> db.movies.find({  
  year: 1932,  
  type: "movie",  
  $or: [ { genres: "Drama" }, { languages: "French" } ]  
},  
{title: 1, year:1, genres:1, languages: 1}  
)  
< 18
```

### Exercise 3.

All documents of [USA movies](#) with between [5 and 9 awards](#) that were produced [between 2012 and 2014](#).

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

```
> db.movies.find({
  countries: "USA" ,
  type: "movie" ,
  awards: { $gte: 5, $lte: 9 } ,
  year: { $gte: 2012, $lte: 2014 }
})
<
> db.movies.find({
  countries: "USA" ,
  type: "movie" ,
  "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, h
  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 acti
  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',
    'Icelandic'
  ],
}
```

## Level 2.

### Exercise 1.

How many comments a user writes who uses "[GAMEOFTHRON.ES](#)" as his/her email domain?

```
db.comments.countDocuments(
  { email: { $regex: "@GAMEOFTHRON.ES$", $options: "i" } }
)
```

```
> db.comments.countDocuments({ email: { $regex: "@GAMEOFTHRON.ES$", $options: "i" } })
< 22841
```

## 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", theater_count: { $sum: 1 } }
  },
  { $sort: { theater_count: -1 } }
])
```

```
> db.theaters.aggregate([
  { $match: {"location.address.state": "DC"} },
  { $group:
    { _id: "$location.address.zipcode", theater_count: { $sum: 1 } }
  },
  { $sort: { theater_count: -1 } }
])
< {
  _id: '20016',
  theater_count: 1
}
{
  _id: '20010',
  theater_count: 1
}
{
  _id: '20002',
  theater_count: 1
}
```

## Level 3.

### Exercise 1.

All [movies](#) 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 }
})
```

```
> db.movies.find( {
  type: "movie",
  directors: "John Landis",
  "imdb.rating": { $gte: 7.5, $lte: 8 }
})
< {
  _id: ObjectId('573a1397f29313caabce6d94'),
  fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a s
  imdb: {
    rating: 7.6,
    votes: 84834,
    id: 77975
  },
  year: 1978,
  plot: 'At a 1962 college, Dean Vernon Wormer is determined to expel the entire Delta Tau Ch
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
    'Italian'
  ],
  writers: [
    'Harold Ramis',
    'Douglas Kenney',
    'Chris Miller'
  ],
  type: 'movie',
  tomatoes: {
    website: 'http://www.animalhouse.com/',
    viewer: {
```

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

## Exercise 2.

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

