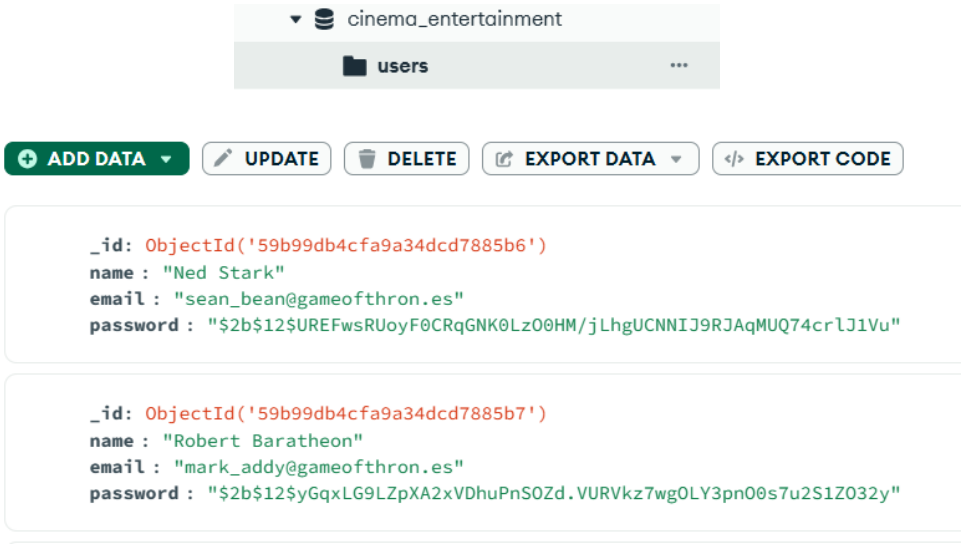
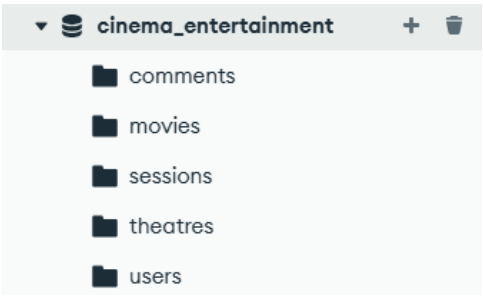
Level 1

Create a database using MongoDB, using the attached files as collections.

Once I have MongoDB installed, I create a database called **cinema_enterteinment** with the first collection called users and add the data from the .json file.



I repeat the same steps for the rest of the collections: theatres, sessions, movies and comments, and I obtain the following structure:



Exercise 1

1. Show the first 2 comments in the database.

```
>_MONGOSH
> use cinema_entertainment
< switched to db cinema_entertainment
> db["comments"].find({}).sort({_id:1}).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
    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. D
    date: 1987-02-10T00:29:36.000Z
  }
```

2. How many users do we have registered?

```
> db["users"].find().count()
< 185
```

3. How many cinemas are there in the state of California?

```
> db.theatres.find({"location.address.state": "CA"}).count()
< 169
```

4. Who was the first user to register?

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

5. How many comedy movies are there in our database?

```
> db.movies.find({genres: "Comedy"}).count()
< 7024
```

- I've executed MongoDB queries on the corresponding collections (users, theatres, movies, comments, sessions), as per the instructions.
- I've applied filters by fields and conditions to obtain the requested records.
- I've used sort and limit, when the exercise required first records.

Exercise 2

Show me all documents for movies produced in 1932, but where the genre is drama or the language is French.

```
> db.movies.find({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 starts talking.
    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/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_
    title: 'The Blood of a Poet',
    lastupdated: '2015-09-16 13:13:05.537000000',
    languages: [
      'French'
    ],
    released: 2010-05-20T00:00:00.000Z,
    directors: [
      'Jean Cocteau'
```

- I've executed a query on the movies collection.
- I've filtered by year: 1932. I've applied an $or condition to include movies with genres: "Drama" or languages: "French".
- I've obtained as a result all documents that meet at least one of these two conditions.

Exercise 3

Show me all documents for American movies with between 5 and 9 awards that were produced between 2012 and 2014.

```
> 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 years for the ma
    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 embarking on a global j
    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'
    ],
```

- I've executed a query on the movies collection to obtain American films.
- I've filtered by a year range between 2012 and 2014, and I filtered by the number of awards to keep only those with between 5 and 9.
- I've obtained as output the documents that simultaneously meet these conditions.

Level 2

Exercise 1

Count how many comments a user has written who uses "GAMEOFTHRON.ES" as their email domain.

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

- I've obtained the total number of comments written by that domain.

Exercise 2

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

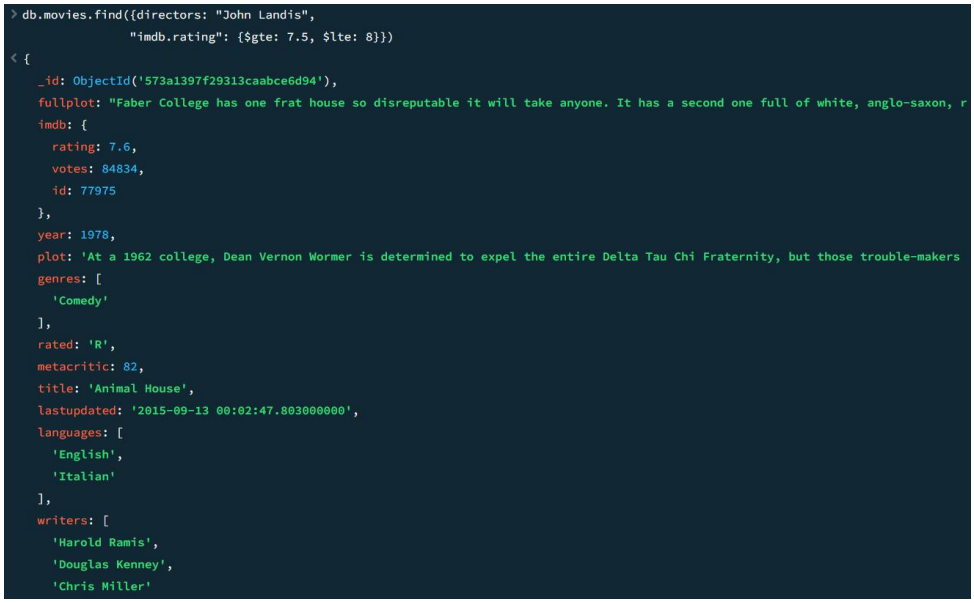
```
> db.theatres.aggregate([
    {$match: {"location.address.state": "DC"}},
    {$group: {_id: "$location.address.zipcode",
              numero_cinemas: {$sum: 1}}},
    {$sort: {numero_cinemas: -1, _id: 1}}
                        ])
< {
    _id: '20002',
    numero_cinemas: 1
  }
  {
    _id: '20010',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

- I've filtered the theatres collection by the state DC.
- I've grouped the documents by location.address.zipcode.
- I've counted how many cinemas there are in each zip code.

Level 3

Exercise 1

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

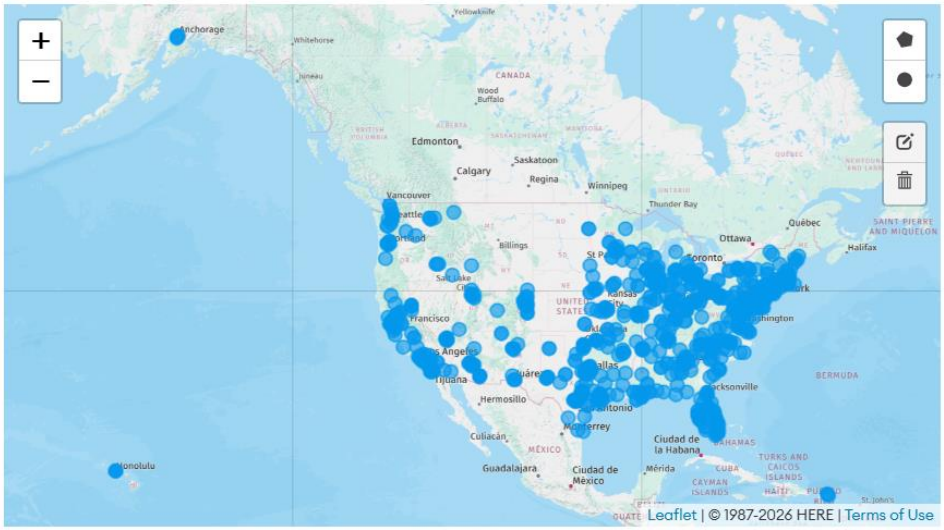


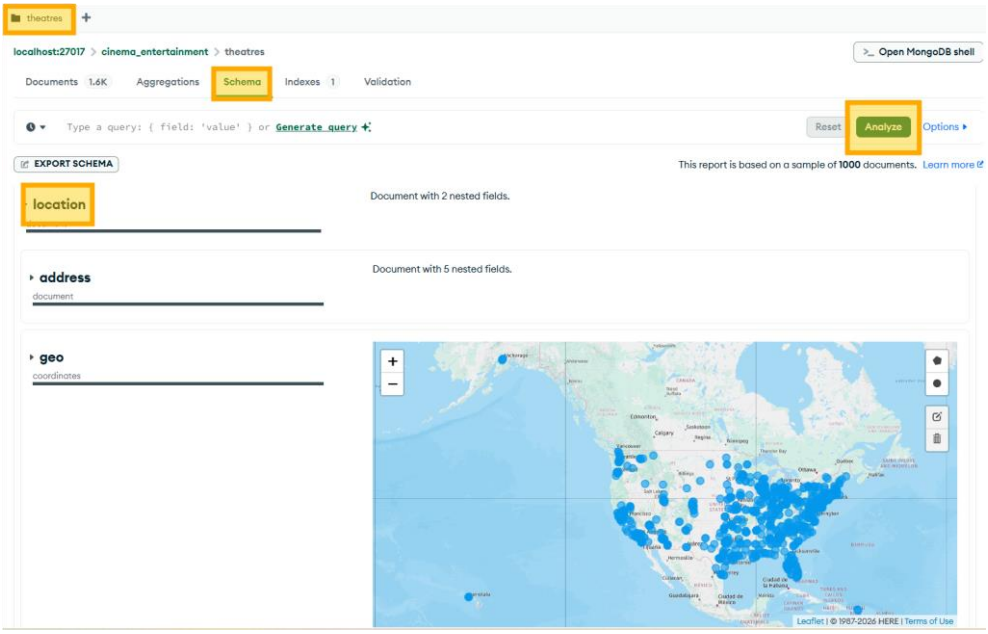
```
> db.movies.find({directors: "John Landis",
                  "imdb.rating": {$gte: 7.5, $lte: 8}})
< {
    _id: ObjectId('573a1397f29313caabce6d94'),
    fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a second one full of white, anglo-saxon, r
    imdb: {
      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 Chi Fraternity, but those trouble-makers
    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'
```

- I've filtered the movies collection by directors: "John Landis".
- I've filtered by a rating range in imdb.rating between 7.5 and 8.
- I've obtained the documents that meet both conditions.

Exercise 2

Display on a map the location of all theatres in the database.





- I've accessed the **theatres** collection within the cinema_entertainment database in MongoDB Compass.
- I've gone to the **Schema** tab and run **Analyze**.
- I've located the **location** field and visualized the points on the integrated map.