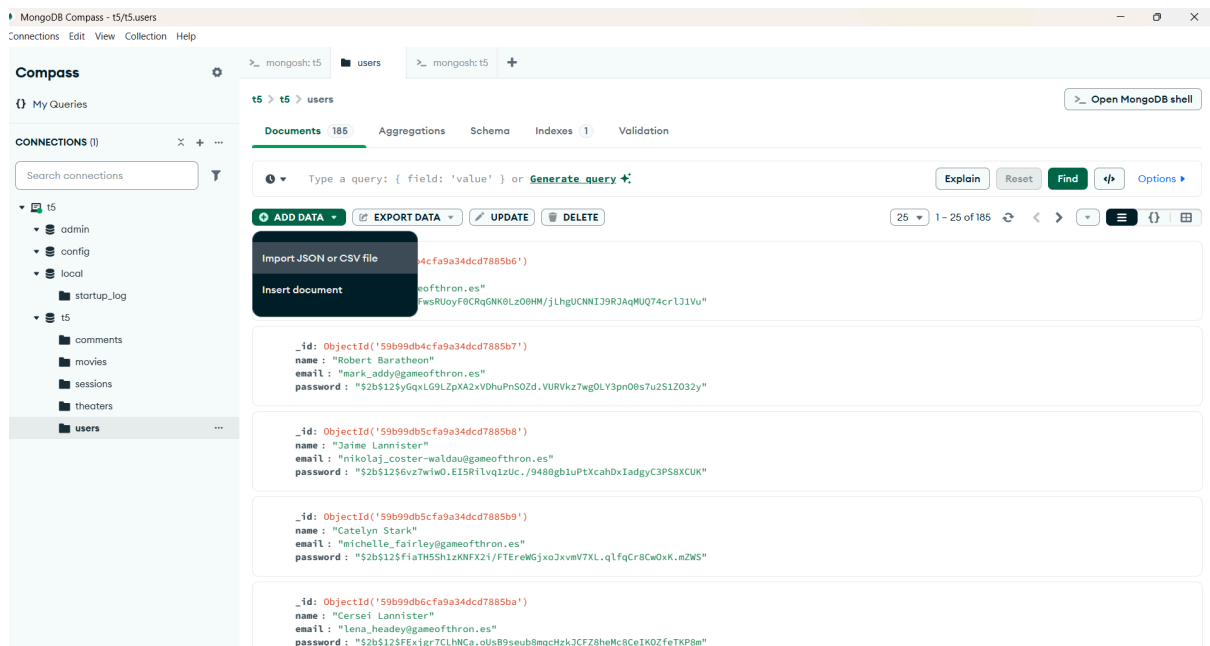


Nivell 1

Después de instalar Mongo DB Compass, abrimos la aplicación. Le damos a crear base de datos (la he llamado t5) y dentro de esta creamos una colección por cada archivo .json (como también especifica el enunciado).

Una vez creado nos aparecerán en el menú de la izquierda como se puede ver en la captura. Luego tenemos que añadir la información que nos da el enunciado (que son los archivos .json). Clicamos sobre “ADD DATA” y vamos importando los archivos para la colección.



Exercici 1

- Mostra els 2 primers comentaris que hi ha en la base de dades.



- Quants usuaris tenim registrats?

```
> db.users.countDocuments()  
< 185
```

- Quants cinemes hi ha en l'estat de Califòrnia?

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

- Quin va ser el primer usuari/ària en registrar-se?

No podemos saberlo ya que no tenemos ningún atributo sobre fecha o fecha de registro. Pero si nos basamos por el id, podemos ver que sería

name: "Ned Stark"

- Quantes pel·lícules de comèdia hi ha en la nostra base de dades?

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

Exercici 2

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel lists several connections, with 't5' selected. The main panel displays a MongoDB query in the 'MONGOSH' terminal:

```
> db.movies.find({  
  year: 1932,  
  $or: [  
    { genres: "Drama" },  
    { languages: "French" }  
  ]  
})  
< {  
  _id: ObjectId('573a1391f2913caabed9458'),  
  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. The artist tries to',  
  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/H/MV5BYWY3ODk4ZmYtMjY0YTYxXkFqcGdeQXVyNzI4MDMyNTU@._V1_S1600_SX677_AL_.jpg',  
  title: 'The Blood of a Poet',  
  lastupdated: '2015-09-16 13:13:05.537000000',  
  languages: [  
    'French'  
  ],  
  released: 2010-05-26T00:00:00.000Z,  
  directors: [  
    'Jean Cocteau'  
  ],  
}
```

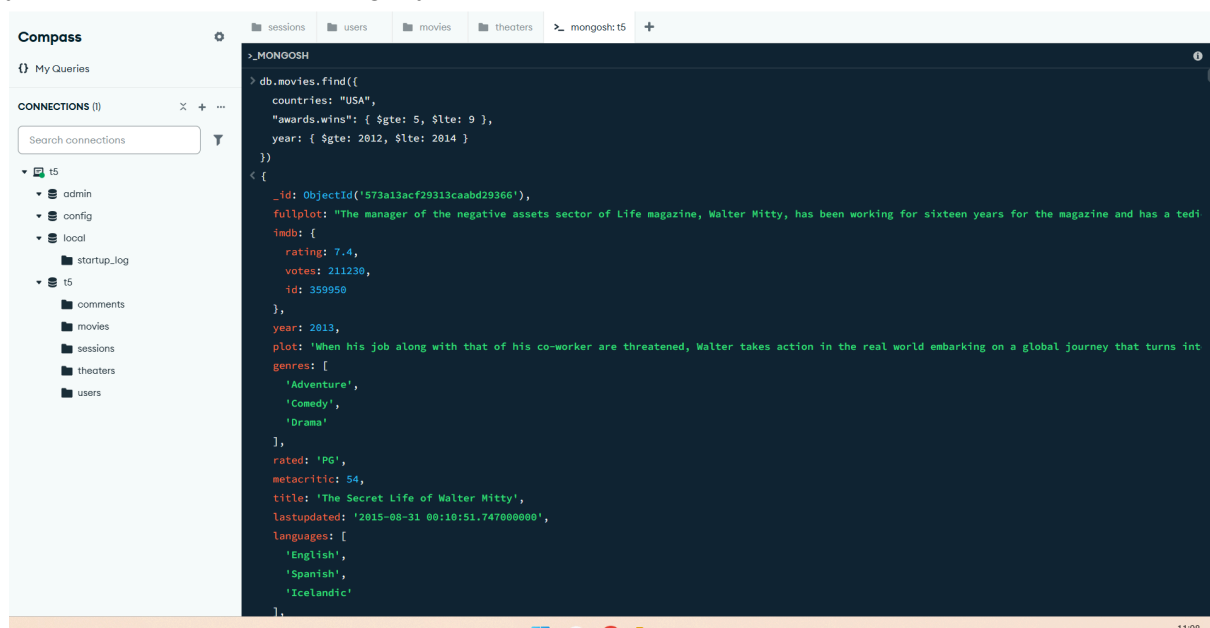
Buscamos con un `or` para “decir” que queremos que nos den las películas del 1932 con Drama como género o que el idioma sea Francés. Realmente sería igual que buscáramos por un lado películas del 1932 en Francés y por otro lado películas del 1932 de Drama y juntáramos resultados.

Exercici 3

Usamos

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

Como antes hacemos primero un simple “find” para buscar por país, en este caso USA . Luego como nos dan un rango de premios y años, tenemos que usar comparadores. Por ejemplo ponemos que una condición sea que el número de premios sea más grande que 5 y más pequeño que 9 con `$gte` y `$lte` respectivamente.



Nivell 2

Exercici 1

```
> _MONGOSH
```

```
> db.comments.countDocuments({  
  "email": /@gameofthron\.es$/i  
})
```

```
< 22841
```

Aquí buscamos dentro de la colección comments todos los documentos cuyo campo email termina en @gameofthron.es (sin importar mayúsculas/minúsculas gracias al flag i del regex). Luego contamos cuántos hay con el countDocuments().

Exercici 2

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

Primero filtramos (con \$match) solo los cines que están en el estado DC. Luego agrupamos (con \$group) por código postal (zipcode) y contamos cuántos hay. Así nos sale el código postal y el total de cines por código postal.

Nivell 3

Exercici 1

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

Como hicimos antes, filtramos por el director que nos piden y con comparadores ajustamos para que la puntuación sea más grande de 7,5 y menos que 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, rich young men who a
  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 have other plans fo
  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',
```

Exercici 2

Para ver el mapa de los cines, vamos a Schema, le damos a Analyse y buscamos el atributo de coordenadas para abrir el mapa como se puede ver en la captura.

Type a query: { field: 'value' } or [Generate query](#)

Reset

Analyze

Options

EXPORT SCHEMA

This report is based on a sample of 1000 documents. [Learn more](#)

document

geo

coordinates

