

Tasca S5.01. Consultes amb MongoDB

•Descripció

•Treballarem amb una base de dades que conté col·leccions relacionades amb una aplicació d'entreteniment cinematogràfic:

- users: Emmagatzema informació d'usuaris/es, incloent-hi noms, emails i contrasenyes xifrades.
- theatres: Conté dades de cinemes, com ID, ubicació (direcció i coordenades geogràfiques).
- sessions: Guarda sessions d'usuari, incloent-hi ID d'usuari i tokens JWT per a l'autenticació.
- movies: Inclou detalls de pel·lícules, com a trama, gèneres, durada, elenc, comentaris, any de llançament, directors, classificació i premis.
- comments: Emmagatzema comentaris d'usuaris/es sobre pel·lícules, amb informació de l'autor/a del comentari, ID de la pel·lícula, text del comentari i la data.

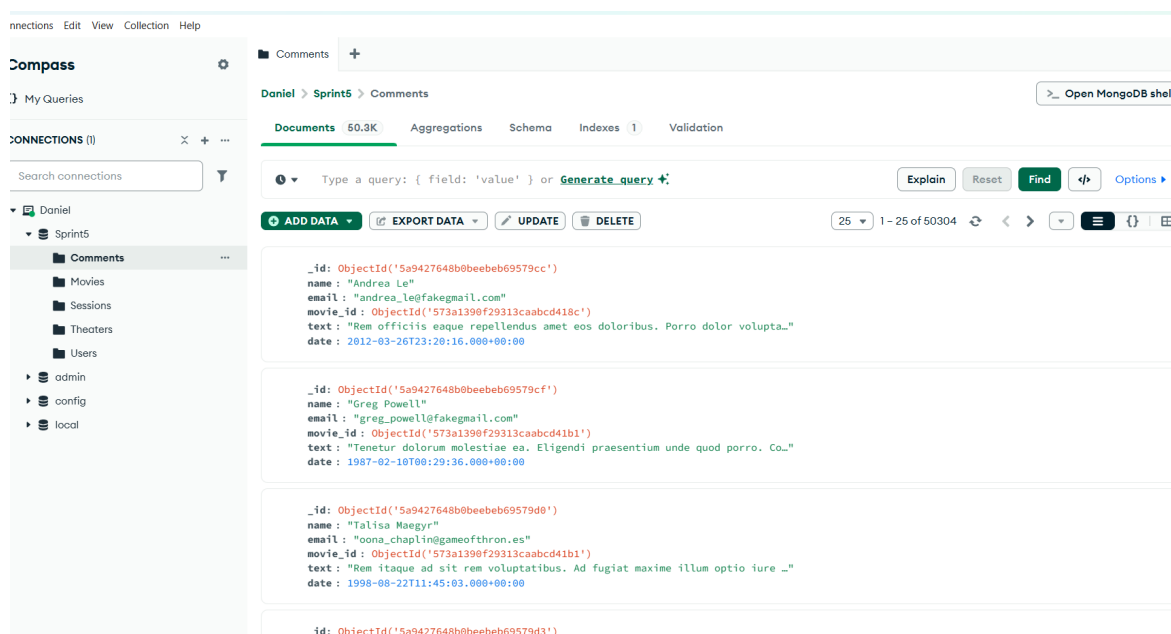
•Duràs a terme algunes consultes que et demana el client/a, el qual està mesurant si seràs capaç o no de fer-te càrrec de la part analítica del projecte vinculat amb la seva base de dades.

•Nivell 1

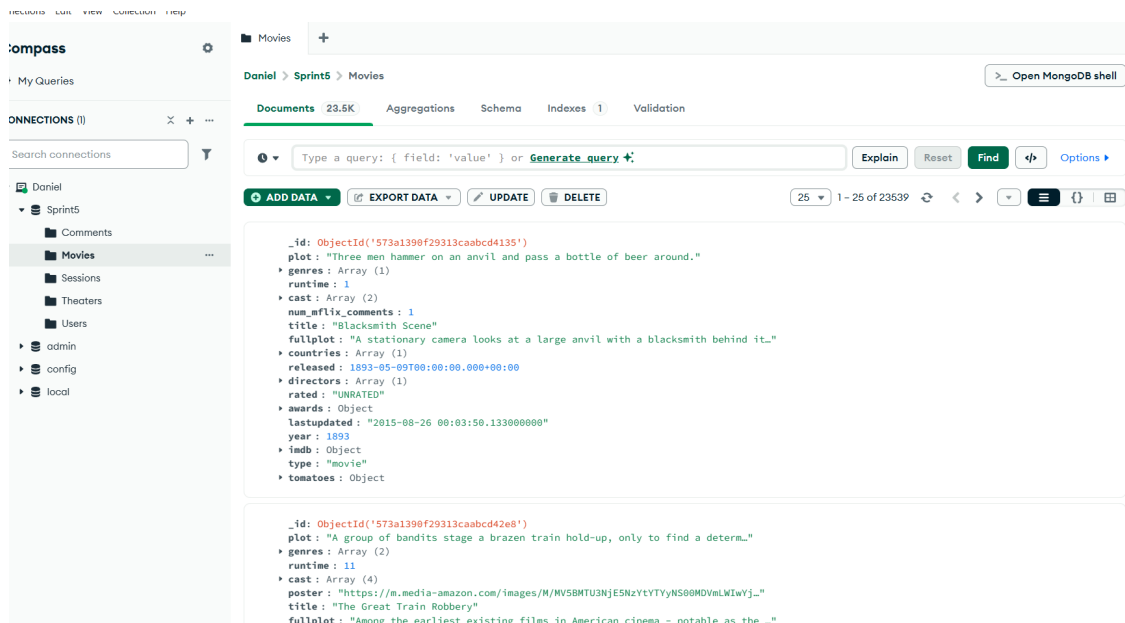
•Crea una base de dades amb MongoDB utilitzant com a col·leccions els arxius adjunts.

- Primero creamos la base de datos , en este caso le hemos puesto el nombre Sprint5 y cargamos las collecciones de los archivos JSON que se nos da a final del ejercicio.

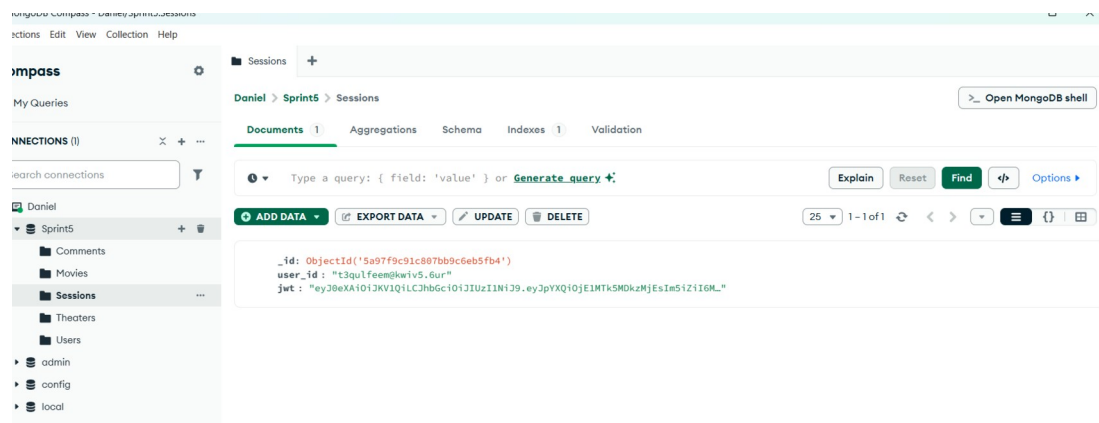
- Cargamos la colección Comments.



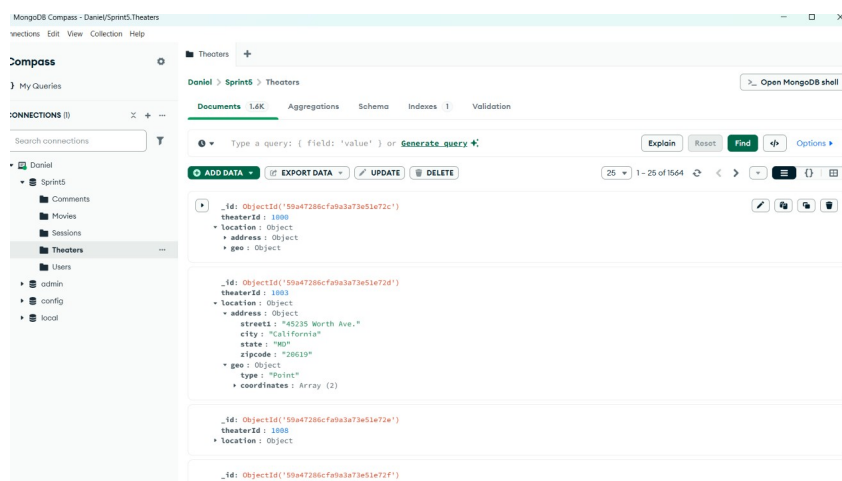
- Cargamos la colección Movies.



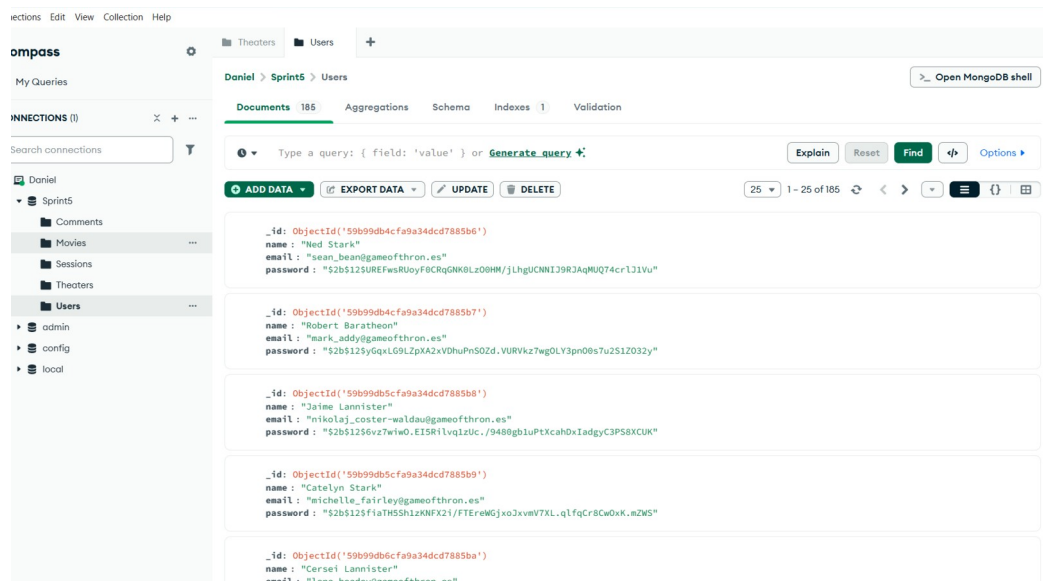
- Cargamos la colección Sessions.



- Cargamos la colección Theaters.



- Cargamos la colección Users.



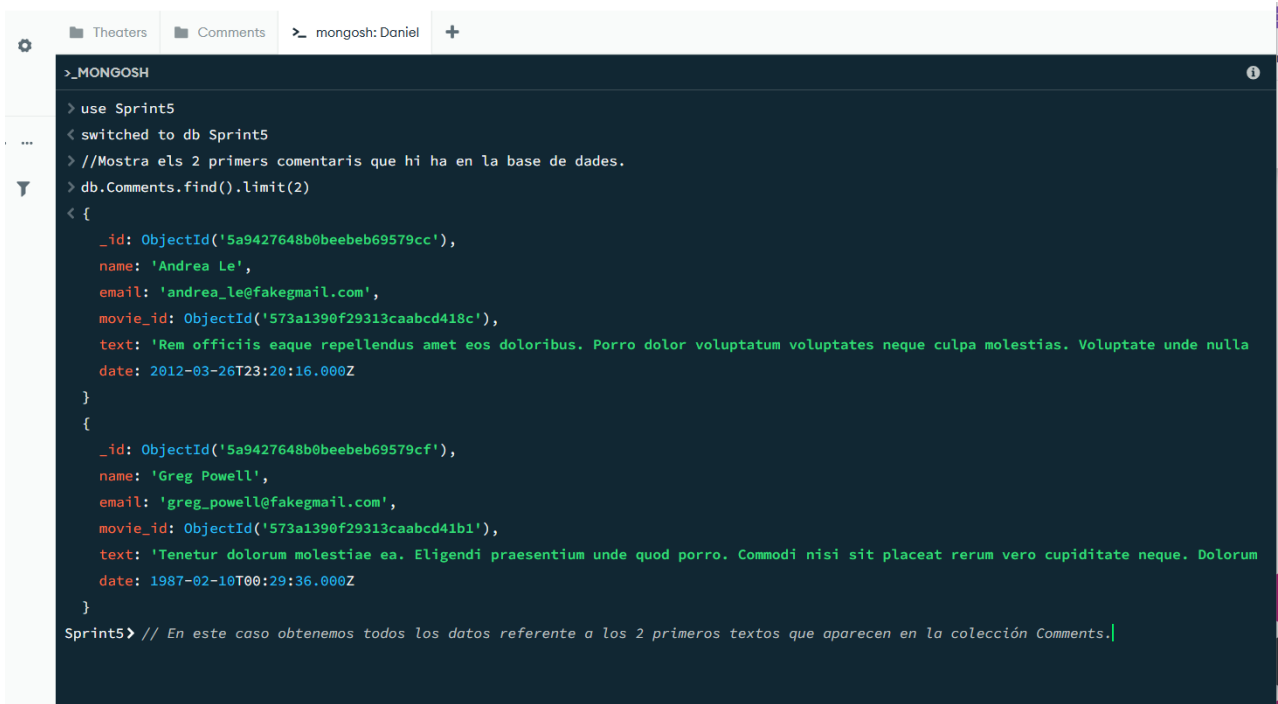
- Una vez cargada la base de datos en MongoDB abriremos MongoSH para realizar las siguientes consultas.

• Ejercicio 1

- Muestra los 2 primeros comentarios que hay en la base de datos.

- Realizamos la consulta en MONGODB con el siguiente código.

- Con el find() buscamos todos los registros de la colección y mediante el limit(2) solo nos devolverá los 2 primeros que aparecen en la colección de Comments.



- Quants usuaris tenim registrats?

- En esta ocasi3n usamos la colecci3n Users y con el CountDocuments() obtenemos el resultatdo que en esta ocasi3n son 185.



```
>_MONGOSH
> use Sprint5
< switched to db Sprint5
> // Quants usuaris tenim registrats?
> db.Users.countDocuments()
< 185
Sprint5>
```

- Quants cinemes hi ha en l'estat de Calif3rnia?

- En esta ocasi3n relaizamos la consulta mediante la colecci3n Theaters con un CountDocuments({"location.address.state":"CA"}) para obtener el numero de cines que se encuentran en el estado de California.

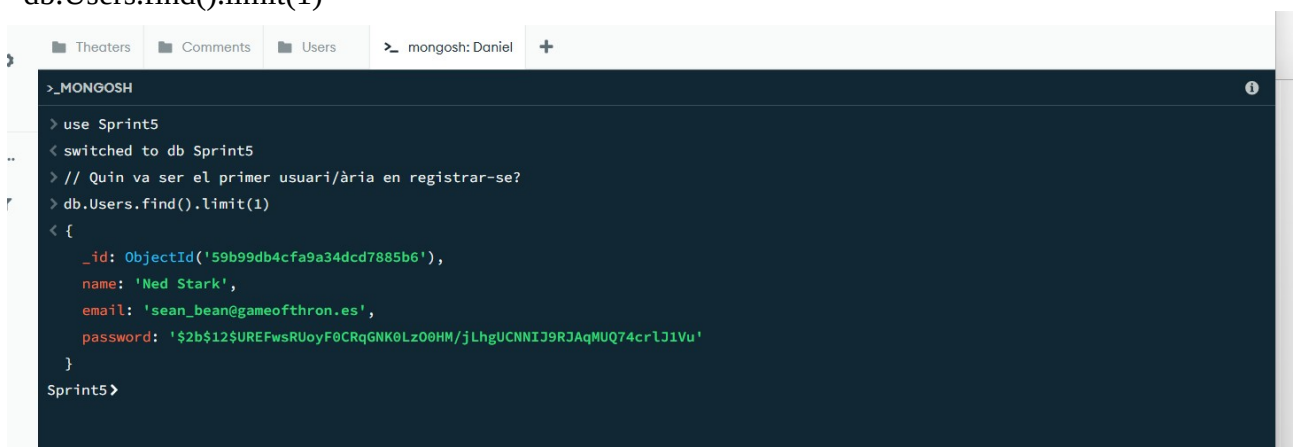


```
>_MONGOSH
> use Sprint5
< switched to db Sprint5
> // Quants cinemes hi ha en l'estat de Calif3rnia?
> db.Theaters.countDocuments({"location.address.state":"CA"})
< 169
Sprint5>
```

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

- En este caso realizamos la siguiente consulta:

db.Users.find().limit(1)



```
>_MONGOSH
> use Sprint5
< switched to db Sprint5
> // Quin va ser el primer usuari/ària en registrar-se?
> db.Users.find().limit(1)
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF9CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crLJ1Vu'
}
Sprint5>
```

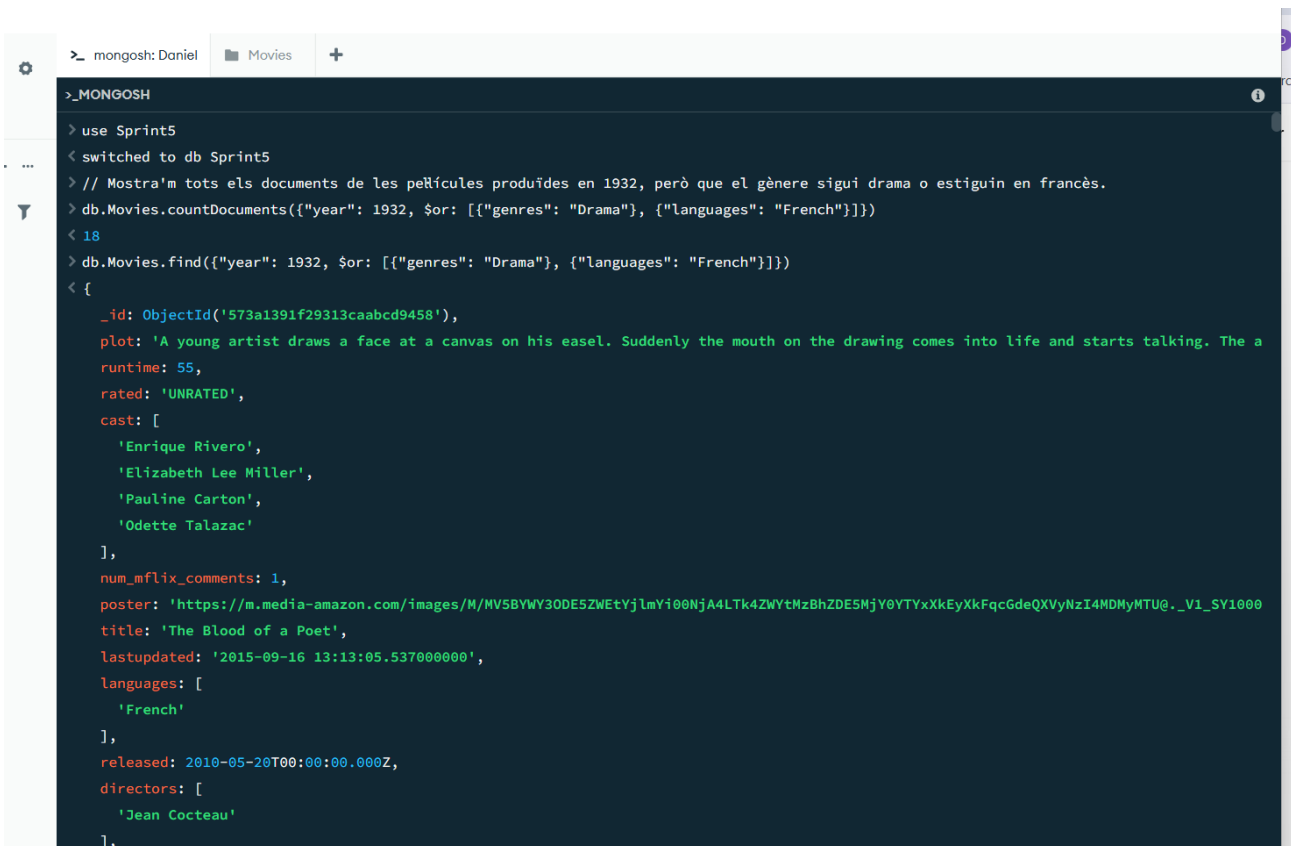
- Quantes pel·lícules de comèdia hi ha en la nostra base de dades?
- En este caso usamos el siguiente código:
db.Movies.CountDocuments({"genres": "Comedy"}) y obtenemos el resultado de 7024 películas que son comedia.



```
>_MONGOSH
> use Sprint5
< switched to db Sprint5
> // Quantes pel·lícules de comèdia hi ha en la nostra base de dades?
> db.Movies.countDocuments({"genres": "comedy"})
< 0
> db.Movies.countDocuments({"genres": "Comedy"})
< 7024
Sprint5>
```

·Exercici 2

- Mostra'm tots els documents de les pel·lícules produïdes en 1932, però que el gènere sigui drama o estiguin en francès.
- En esta ocasión hemos realizamos dos consultas , la primera un conteo de las que cumplen las condiciones de ser del año 1932 que tengan genero Drama o lenguaje en Frances y la consulta nos devuelve 18 esto lo hemos realizado con un countDocuments. En la 2ªconsulta le hemos puesto el mismo código pero con find para que nos devuelva todos los valores de la consulta que cumplan esas condiciones.

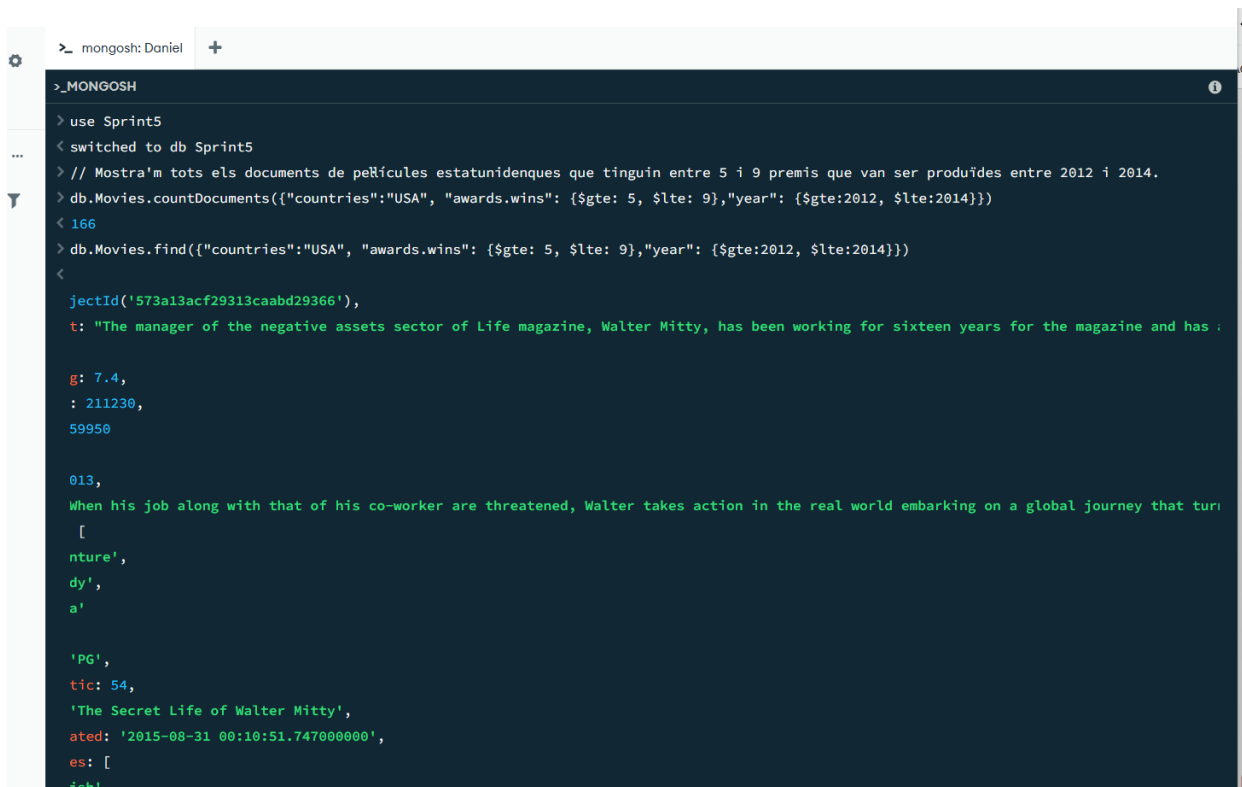


```
>_MONGOSH
> use Sprint5
< switched to db Sprint5
> // Mostra'm tots els documents de les pel·lícules produïdes en 1932, però que el gènere sigui drama o estiguin en francès.
> db.Movies.countDocuments({"year": 1932, $or: [{"genres": "Drama"}, {"languages": "French"}]})
< 18
> db.Movies.find({"year": 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. The a
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/MV5BYWY3ODE5ZWYjLmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTxxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_S1000
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'
],
}
```

•Exercici 3

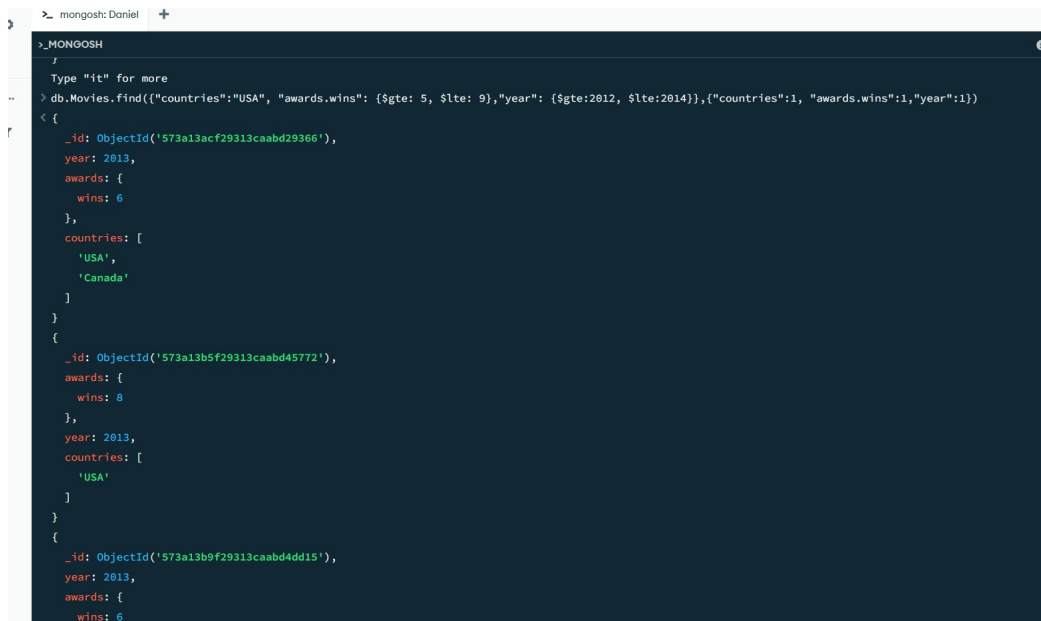
•Mostra'm tots els documents de pel·lícules estatunidenques que tinguin entre 5 i 9 premis que van ser produïdes entre 2012 i 2014.

- Hemos realizado la misma operación que en el ejercicio anterior primero un countDocuments para ver cuantos documentos de la colección cumplen con estos requisitos en esta ocasión utilizamos \$gte que indica valores iguales o superiores al 5 que es el numero de premios y al año en este caso 2012 el \$lte signifiva que sean igual o menores (seria como realizar un between de sql)



```
> mongosh: Daniel +
> _MONGOSH
> use Sprint5
< switched to db Sprint5
> // Mostra'm tots els documents de pel·lícules estatunidenques que tinguin entre 5 i 9 premis que van ser produïdes entre 2012 i 2014.
> db.Movies.countDocuments({"countries":"USA", "awards.wins": {$gte: 5, $lte: 9}, "year": {$gte:2012, $lte:2014}})
< 166
> db.Movies.find({"countries":"USA", "awards.wins": {$gte: 5, $lte: 9}, "year": {$gte:2012, $lte:2014}})
<
  {
    "_id": "573a13acf29313caabd29366",
    "title": "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for the magazine and has :
    "genre": "g: 7.4,
    "year": 211230,
    "awards": 59950
    "year": 013,
    "description": "When his job along with that of his co-worker are threatened, Walter takes action in the real world embarking on a global journey that turn
    "cast": [
      "nature",
      "dy",
      "a"
    ],
    "rating": "PG",
    "duration": "tic: 54,
    "title": "'The Secret Life of Walter Mitty'",
    "released": "ated: '2015-08-31 00:10:51.747000000'",
    "genres": "es: [
    "style": "ish",
  }
```

- En este pantallazo hemos agregado el 2º campo que seria la proyección y le hemos puesto los valores a calcular para comprobar que la consulta es correcta.



```
> mongosh: Daniel
> _MONGODB
Type "it" for more
> db.Movies.find({"countries":"USA", "awards.wins": {$gte: 5, $lte: 9}, "year": {$gte: 2012, $lte: 2014}}, {"countries":1, "awards.wins":1, "year":1})
< {
  _id: ObjectId('573a13acf29313caabd29366'),
  year: 2013,
  awards: {
    wins: 6
  },
  countries: [
    'USA',
    'Canada'
  ]
}
{
  _id: ObjectId('573a13b5f29313caabd45772'),
  awards: {
    wins: 8
  },
  year: 2013,
  countries: [
    'USA'
  ]
}
{
  _id: ObjectId('573a13b9f29313caabd4dd15'),
  year: 2013,
  awards: {
    wins: 6
  },
  countries: [
    'USA'
  ]
}
```

•Nivell 2

•Exercici 1

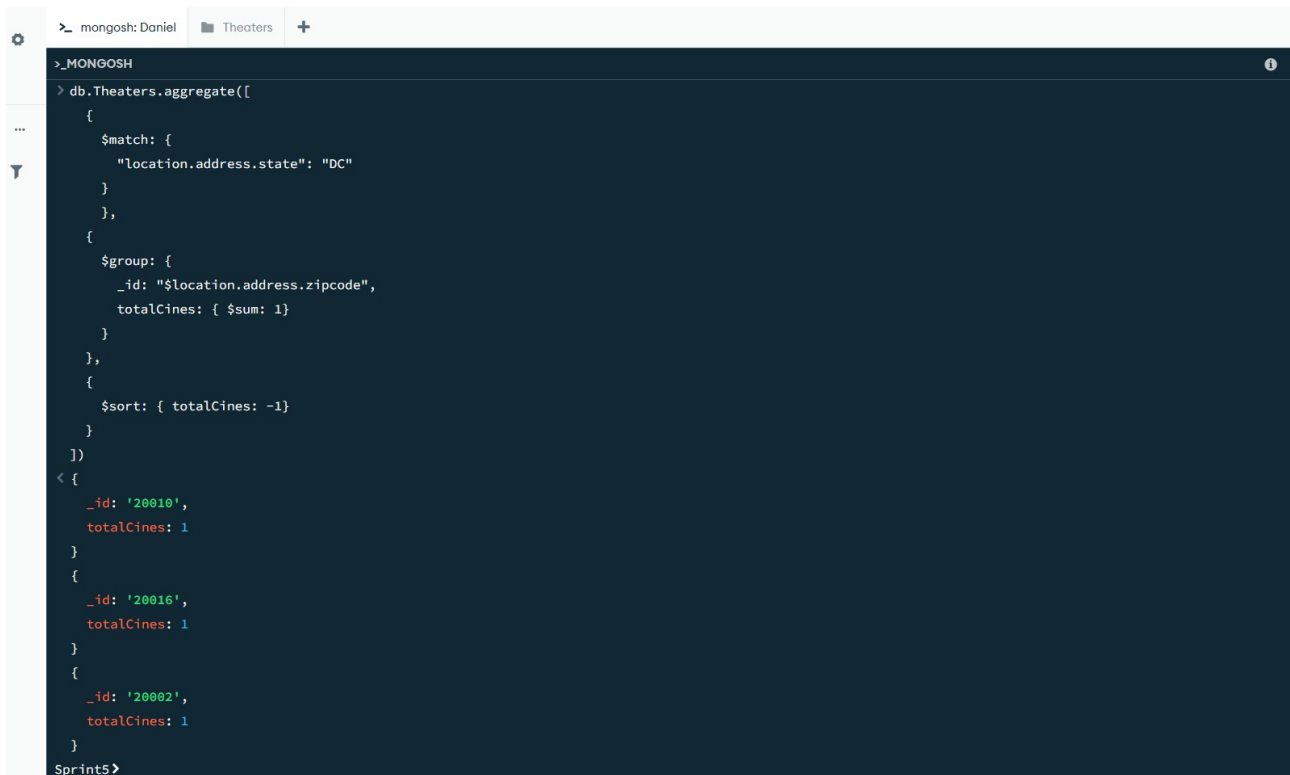
•Compte quants comentaris escriu un usuari/ària que utilitza "GAMEOFTHRON.ES" com a domini de correu electrònic.

- En este ejercicio realizamos una consulta de agregación que procesan los documentos de una colección pasando por distintas etapas, estas etapas constituyen una ruta que se llama Pipeline y la salida de una etapa supondra la entrada en la siguiente. En el código realizado la etapa \$match nos filtra los correos que terminan en gameofthron independientemente esten en mayusculas o en minusculas el \$group nos agrupa por email y con el \$sum:1 le indicamos que cada correo que sea igual nos sume uno cada vez que lo recorre y finalmente con el \$sort ordenamos el total de comentarios realizado por el mismo correo en -1 para que nos de un orden descendente.

```
> mongosh: Daniel +
>_MONGOSH
> db.Comments.aggregate([
  {
    $match: {
      "email": { $regex: /@gameofthron\.es$/i }
    }
  },
  {
    $group: {
      _id: "$email",
      totalComentarios: { $sum: 1 }
    }
  },
  {
    $sort: { totalComentarios: -1 }
  }
])
< {
  _id: 'roger_ashton-griffiths@gameofthron.es',
  totalComentarios: 331
}
{
  _id: 'nathalie_emmanuel@gameofthron.es',
  totalComentarios: 327
}
{
  _id: 'jonathan_pryce@gameofthron.es',
  totalComentarios: 315
}
```

```
> mongosh: Daniel +
>_MONGOSH
{
  _id: 'jonathan_pryce@gameofthron.es',
  totalComentarios: 315
}
{
  _id: 'sophie_turner@gameofthron.es',
  totalComentarios: 308
}
{
  _id: 'ron_donachie@gameofthron.es',
  totalComentarios: 305
}
{
  _id: 'paul_kaye@gameofthron.es',
  totalComentarios: 304
}
{
  _id: 'gwendoline_christie@gameofthron.es',
  totalComentarios: 302
}
{
  _id: 'anton_lesser@gameofthron.es',
  totalComentarios: 295
}
{
  _id: 'maisie_williams@gameofthron.es',
  totalComentarios: 295
}
```


- Exercici 2
- Quants cinemes hi ha en cada codi postal situats dins de l'estat Washington D. C. (DC)?
- En este ejercicio realizamos también un aggregate donde en la consulta con \$match le pedimos que no de todos los cines que se encuentran en “DC” en el \$group realizamos la agrupación por el código postal y finalmente agrupamos en orden descendiente mediante el \$sort y el resultado son 3 códigos postales con un cine cada uno de ellos



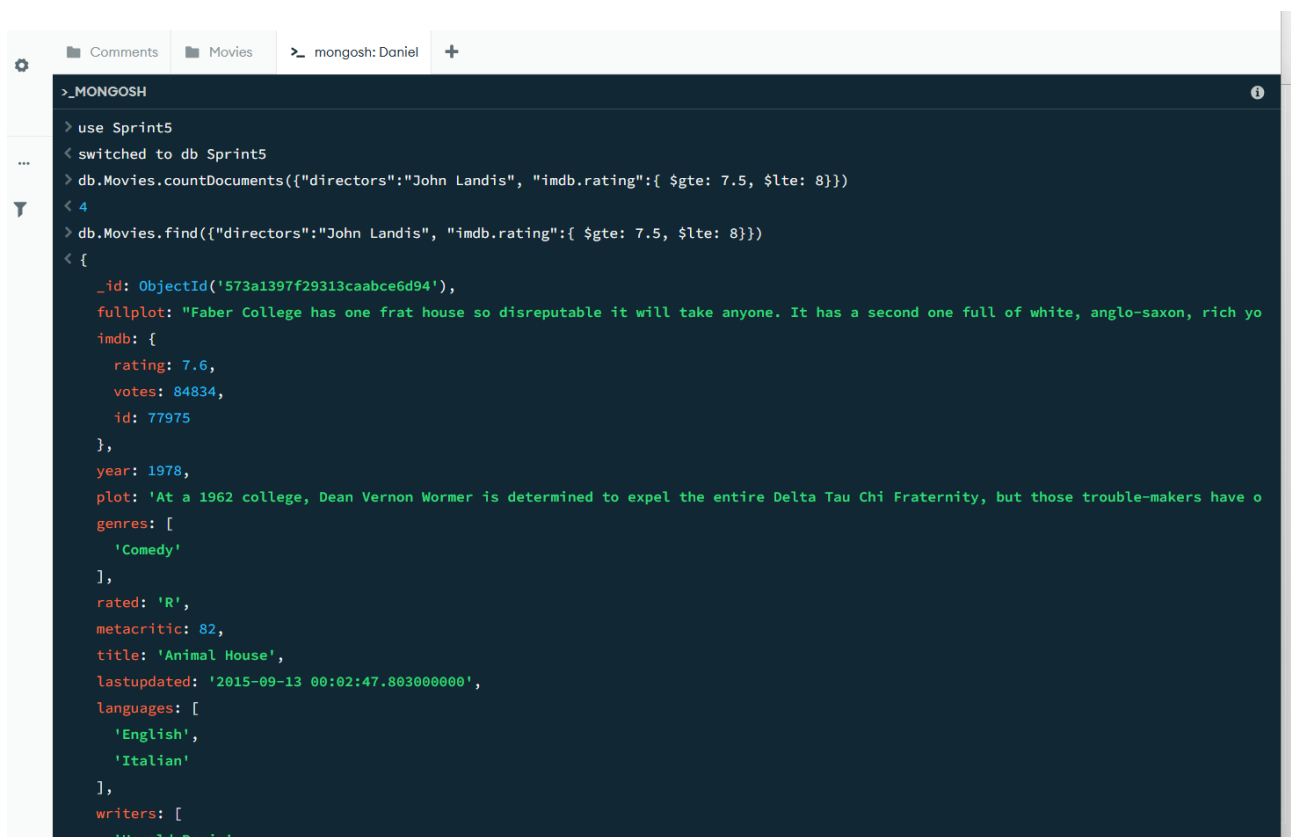
```
> mongosh: Daniel Theaters +
>_MONGOSH
> db.Theaters.aggregate([
  {
    $match: {
      "location.address.state": "DC"
    }
  },
  {
    $group: {
      _id: "$location.address.zipcode",
      totalCines: { $sum: 1 }
    }
  },
  {
    $sort: { totalCines: -1 }
  }
])
< {
  _id: '20010',
  totalCines: 1
}
{
  _id: '20016',
  totalCines: 1
}
{
  _id: '20002',
  totalCines: 1
}
Sprint5>
```

•Nivell 3

•Exercici 1

•Troba totes les pel·lícules dirigides per John Landis amb una puntuació IMDb (Internet Movie Database) d'entre 7,5 i 8.

- En este ejercicio realizamos un find de la colección donde el nombre del director sea John Landin y el imdb.rating este entre 7.5 y 8 y obtenemos el resultado de 4 películas con esas características.



```
> _MONGOSH
> use Sprint5
< switched to db Sprint5
> db.Movies.countDocuments({"directors":"John Landis", "imdb.rating":{" $gte: 7.5, $lte: 8}})
< 4
> 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 yo
  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 o
  genres: [
    'Comedy'
  ],
  rated: 'R',
  metacritic: 82,
  title: 'Animal House',
  lastupdated: '2015-09-13 00:02:47.803000000',
  languages: [
    'English',
    'Italian'
  ],
  writers: [
    'Harold Ramis'
```

•Exercici 2

•Mostra en un mapa la ubicació de tots els teatres de la base de dades.

- En este caso vamos a la colección Theaters a Schema y clicamos en geo y nos aparece el mapa que se nos pide en el ejercicio.

MongoDB Compass - Daniel/Sprint5.Theaters

ions Edit View Collection Help

npass

Comments Movies Theaters mongosh: Daniel

Open MongoDB shell

Daniel > Sprint5 > Theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

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](#)

geo

coordinates

Leaflet | © 1987-2025 HERE | Terms of Use

theaterId

8196 450 2972 8087 446 289 489 550 545 2518 216 2794 8190