

Sprint 9

Apreniendo a usar MongoDB

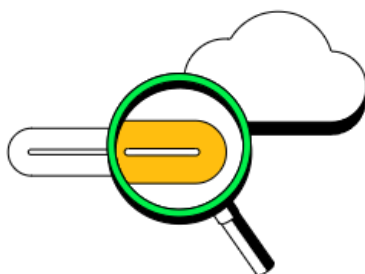
Treballarem amb una base de dades que conté col·leccions relacionades amb una aplicació d'entreteniment cinematogràfic:

- 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.

Conexión

Para crear una nueva conexión he tenido que crear una carpeta en C llamada Data y dentro una subcarpeta llamada db. También he descargado mongosh, después he ejecutado mongod.exe y mongosh.exe y ya, al apretar el botón de *Add new connection* me ha permitido realizar la conexión a localhost.



Welcome to MongoDB Compass

To get started, connect to an existing server or

[+ Add new connection](#)

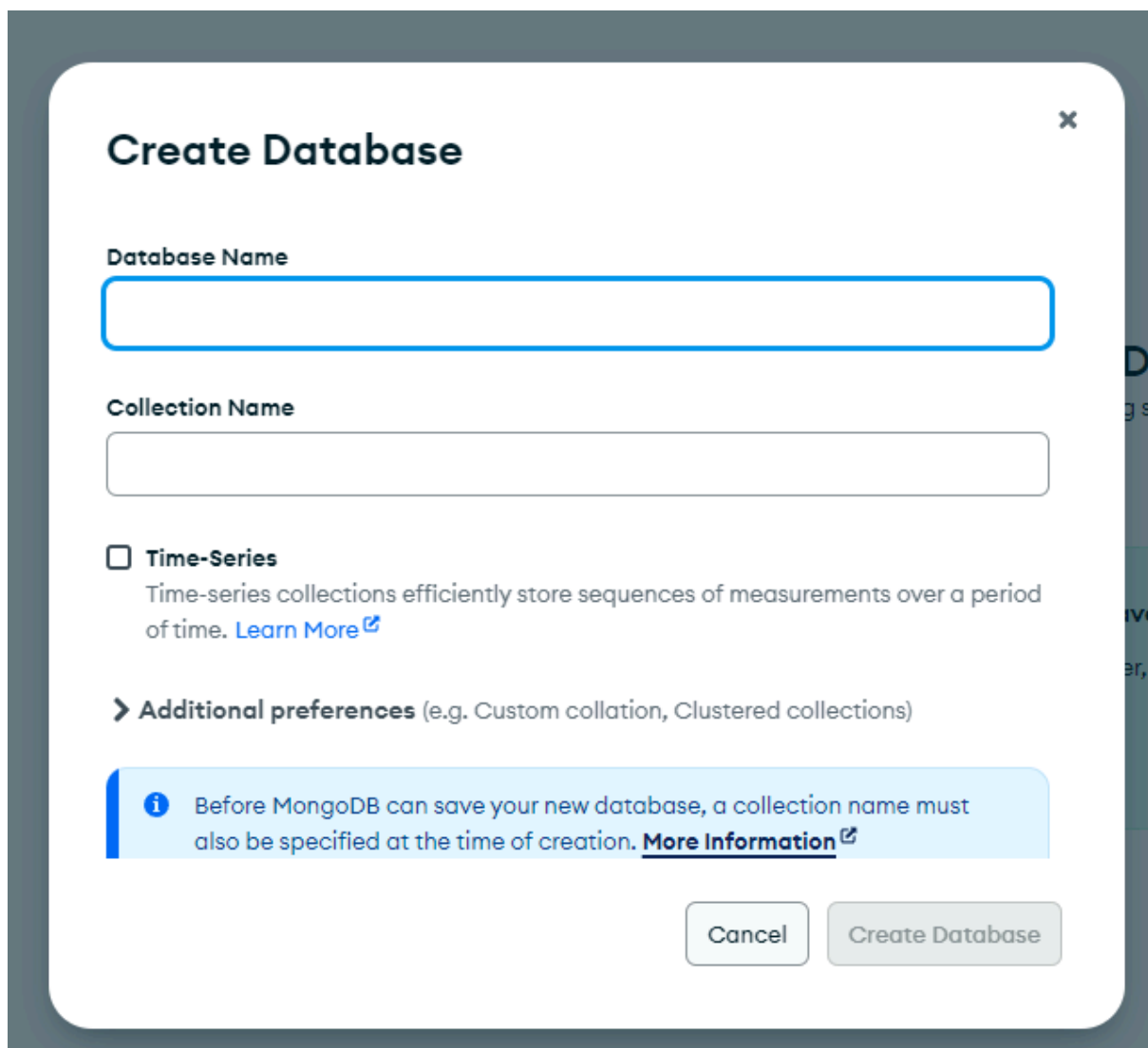
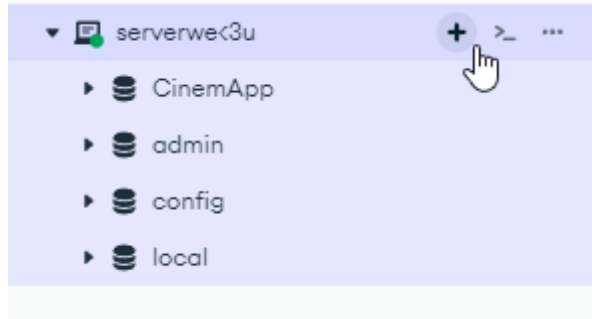
New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#)

[CREATE FREE CLUSTER](#)

Nivell 1

Crea una base de dades amb MongoDB utilitzant com a col·leccions els arxius adjunts. Primero creas la base de datos dándole a la cruz que hay en el lateral del nombre del servidor y rellenando la pantalla que aparece.



Create Database x

Database Name

Collection Name

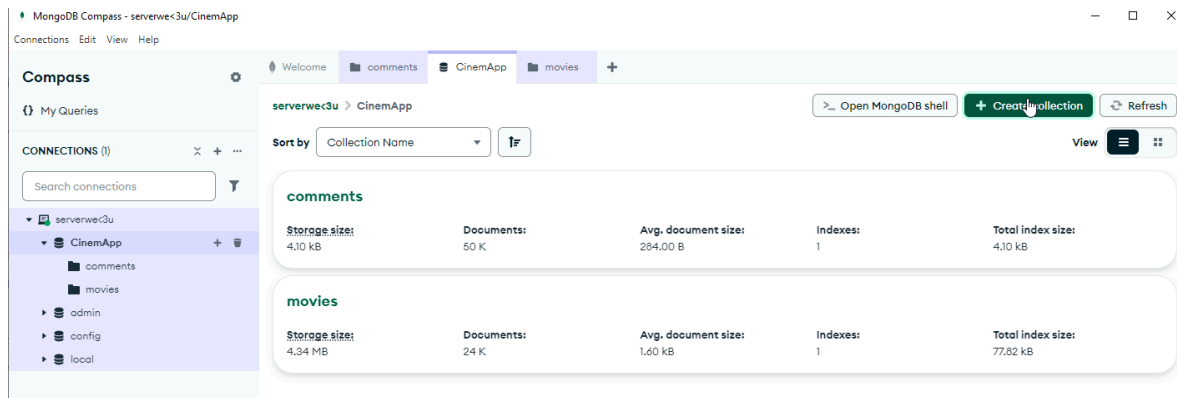
☐ **Time-Series**
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences (e.g. Custom collation, Clustered collections)

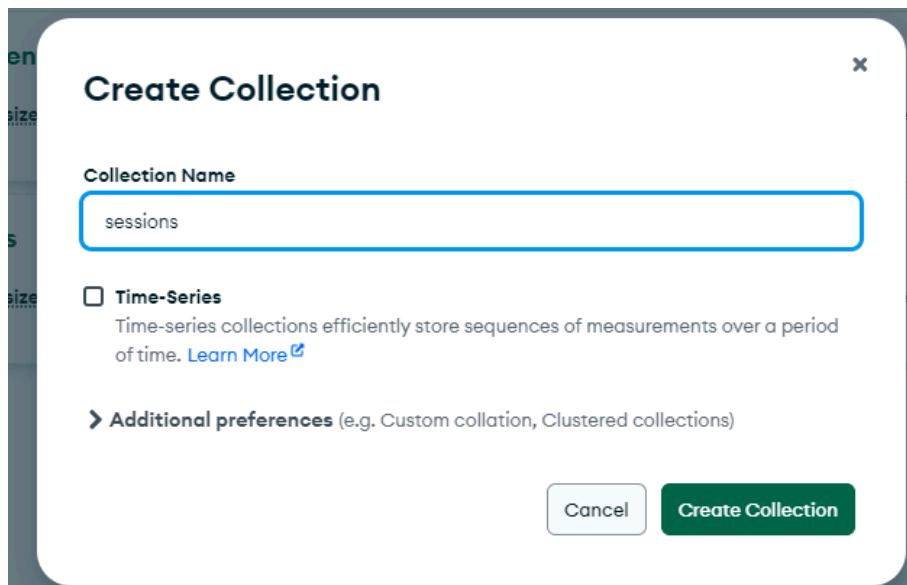
i Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

Cancel Create Database

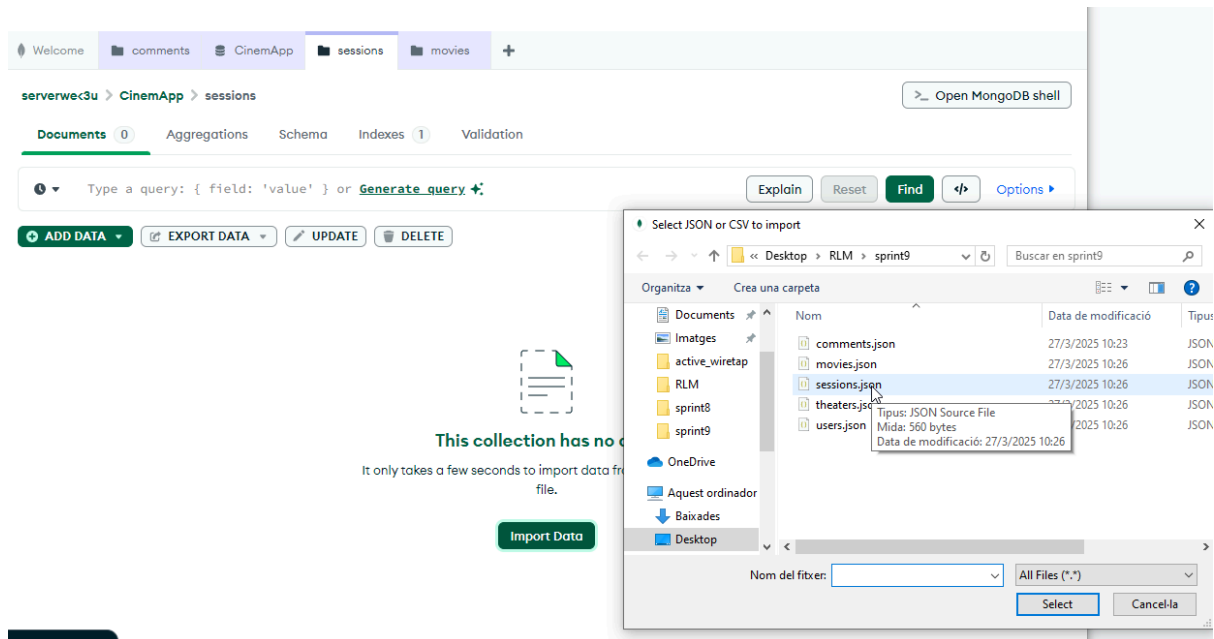
Te apareixerà un símbol de base de dades debaixo del servidor con el nombre elegid per a la misma. Ahí te apareixerà otro símbol de sumar que te permite crear colecciones mediante el botón de + *Create Collection*.

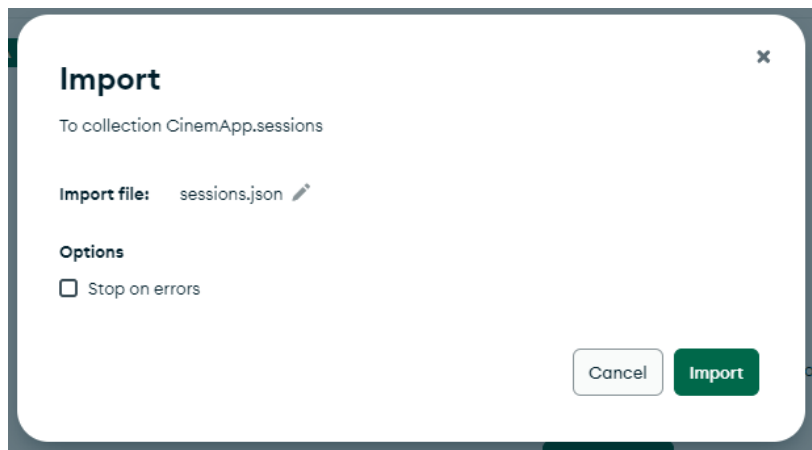


Se abrirá una ventana nueva desde la que escribes el nombre deseado y, opcionalmente, alguna funcionalidad más.

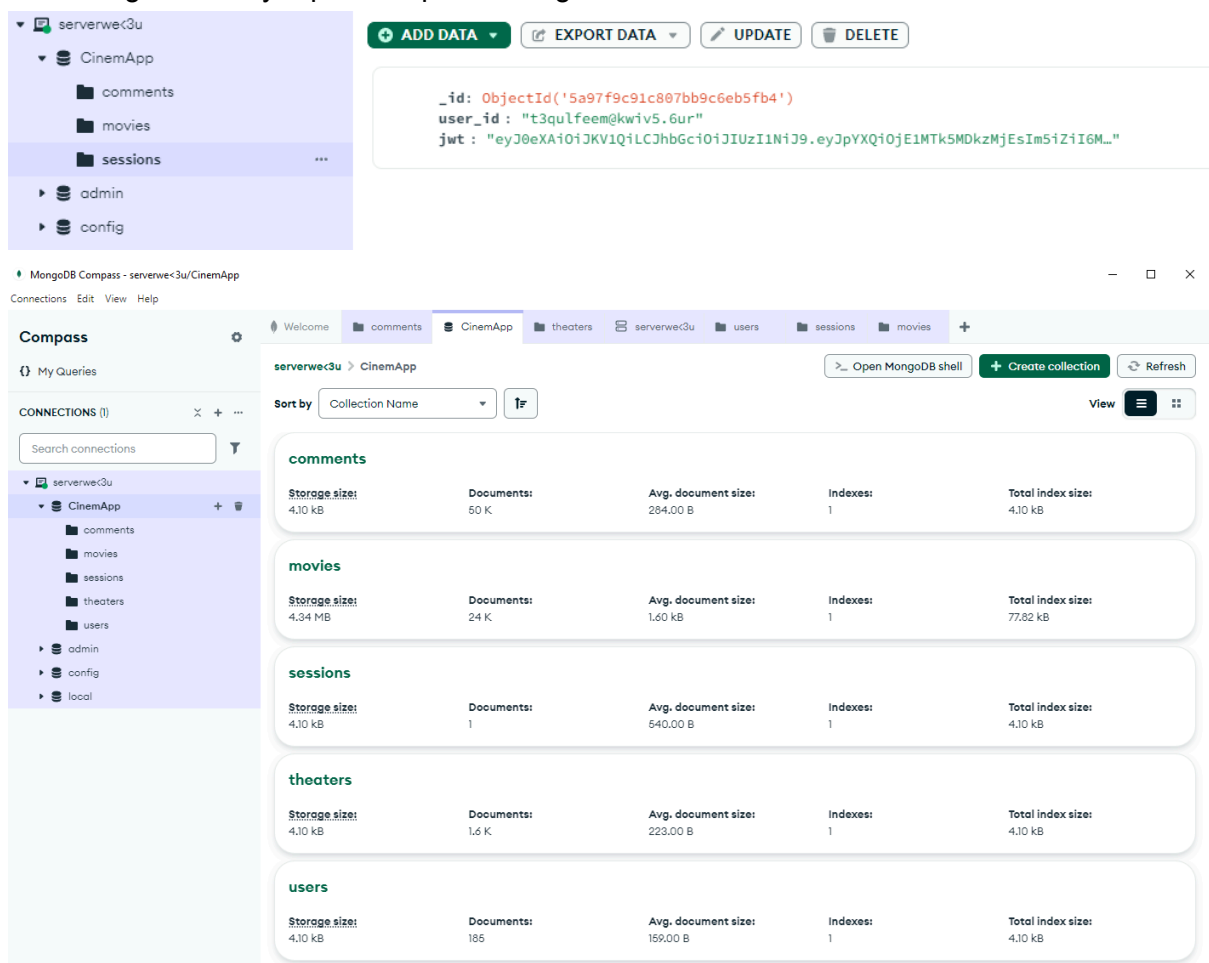


Una vez creada, pulsamos el botón de + **ADD DATA**, seleccionamos la carpeta dónde guardamos el archivo y lo seleccionamos.





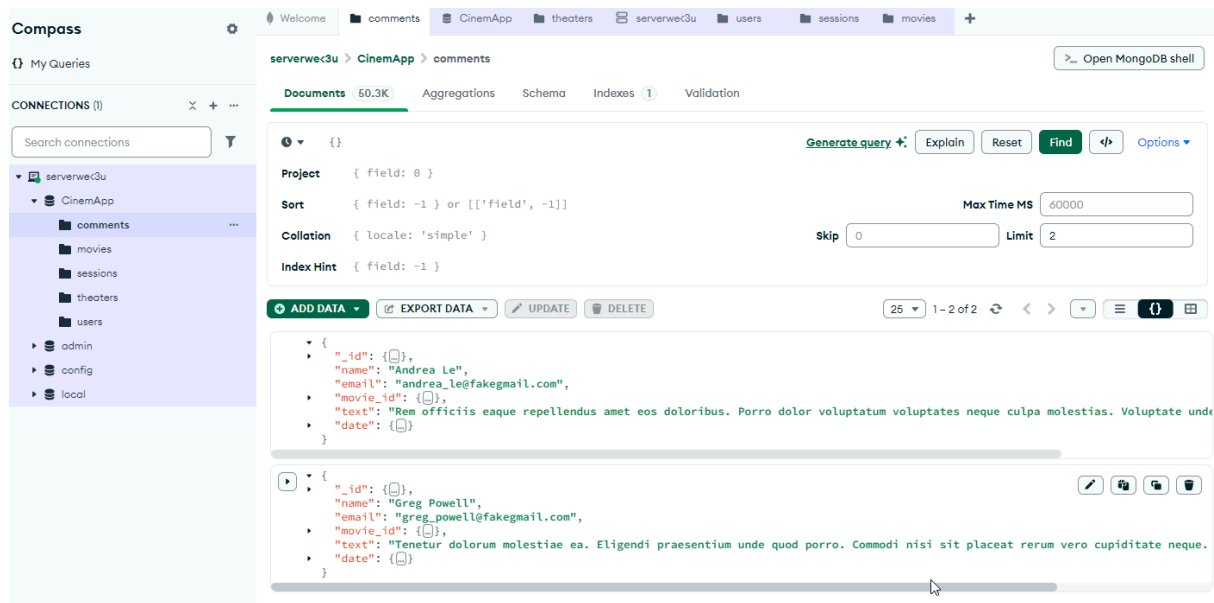
Nos preguntará si queremos importarlo, le damos a *Import* para confirmar. Revisamos que se ha cargado bien y repetimos para las siguientes colecciones.



- Exercici 1

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

Hay dos maneras de hacerlo, la primera, mediante gui. Vas a la base de datos, sección *Documents*, botón *Find*, *Limit 2*.



La segunda manera es a través de Bash, mediante el código que se ve en la captura: `db.comments.find().limit(2)`, es decir, dentro de la base de datos (db) por defecto, y en la colección `comments`, buscará los contenidos (find), reduciendo la búsqueda a 2 (limit(2)).

```
> db.comments.find().limit(2)
< {
  "_id": ObjectId('5a9427648b0beebe69579cc'),
  "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 nulla
  date: 2012-03-26T23:20:16.000Z
}
{
  "_id": ObjectId('5a9427648b0beebe69579cf'),
  "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
  date: 1987-02-10T00:29:36.000Z
}
```

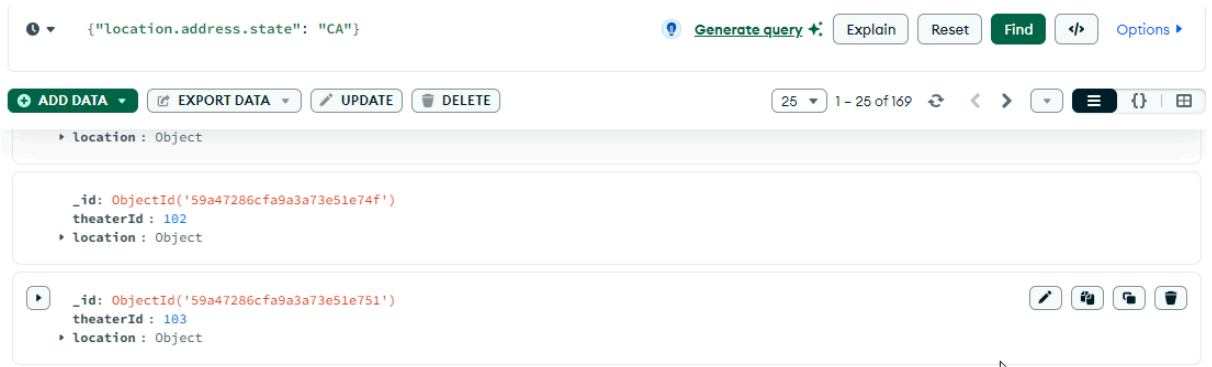
- Quants usuaris tenim registrats?

A través de la gui, sería solo entrar en la colección para ver automáticamente cuántos registros hay. En la Shell usaremos `countDocuments()` para que nos cuente los registros de la colección, que són 185 usuarios en total.

```
> use CinemApp
< switched to db CinemApp
> db.users.countDocuments()
< 185
CinemApp> |
```

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

Lo podemos hacer con una query a través de la gui, dentro de la colección *theaters*.



Con la shell usaremos esa misma query encapsulada en el paréntesis de *countDocuments*.

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

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

El primer usuario en registrarse fue Ned Stark, ya que es el que tiene el primer id, sin embargo si buscamos por fecha y sin límite, veremos que hubo muchos que se registraron el mismo día.

Con *find* busca en todos los documentos, con *sort* ordena los documentos en orden por *date* o *id*, según se prefiera y con *limit* se limita la búsqueda a los primeros de la cantidad deseada, en este caso 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'
}
> db.users.find().sort({date:1}).limit(1)
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu'
}
```

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

Contamos en *movies* aplicando el filtro de género comedia y nos salen 7024.

```
< 0
> db.movies.countDocuments({"genres":"Comedy"})
< 7024
CinemApp> db.movies.countDocuments({"genres":"Comedy"})
```

- Exercici 2

Mostra'm tots els documents de les pel·lícules produïdes en 1932, però que el gènere sigui drama o estiguin en francès.

A través de la gui: En *movies* filtramos por año y luego con *\$or* volvemos a filtrar por género drama o por idioma francés.

The screenshot shows the MongoDB Compass interface. At the top, a query is entered in the 'Find' tab:

```
{
  "year": 1932,
  "$or": [
    { "genres": "Drama" },
    { "languages": "French" }
  ]
}
```

Below the query, there are buttons for 'Generate query', 'Explain', 'Reset', 'Find', and 'Options'. The 'Find' button is highlighted. Below the buttons, there are tabs for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The 'EXPORT DATA' tab is selected. Below the tabs, there are two panels. The top panel shows a summary of the results:

- directors: Array (1)
- writers: Array (1)
- awards: Object
- year: 1932
- imdb: Object
- countries: Array (1)
- type: "movie"
- tomatoes: Object

The bottom panel shows the details of the first result:

```
{
  "_id": ObjectId('573a1392f29313caabcd99a3'),
  "plot": "Junta is hated by the people in the village where she lives, especial...",
  "genres": Array (3),
  "runtime": 85,
  "cast": Array (4),
  "poster": "https://m.media-amazon.com/images/M/MV5BNTQ1NTMzMtQ0DIyYS00MTAxLWE1NT..."
  "title": "The Blue Light",
  "fullplot": "Junta is hated by the people in the village where she lives, especial...",
  "languages": Array (2),
  "released": 1934-05-08T00:00:00.000+00:00,
  "directors": Array (2),
  "writers": Array (2),
  "awards": Object,
  "lastupdated": "2015-08-03 01:02:12.350000000",
  "year": 1932,
  "imdb": Object,
  "countries": Array (1),
  "type": "movie"
}
```

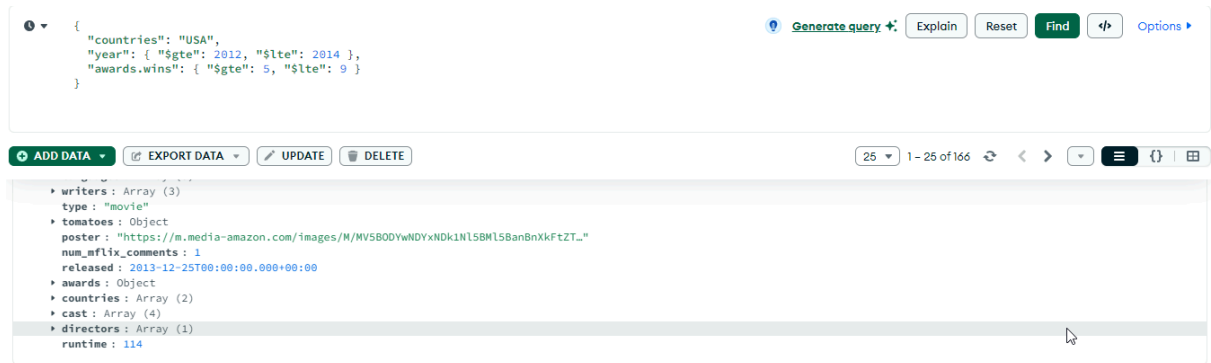
A través del Shell encapsulamos la fórmula anterior dentro del *find*.

```
> 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 artist tries to wipe it away with h
  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/MV5BYWY3ODU5ZWVjYmY1ODNjA4LTk4ZWYtMzBhZDE5MjY0YTUxXkE5XkFqcGdeQXVyNzI4MDMyTU00_V1_SV1000_SX677_AL_.jpg',
  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'
  ],
  writers: [
```

- 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.

A través de la gui nos vamos a *movies* y allí escribimos la query que se ve en la imagen, en la que aplicamos los filtros de país, año y premios. En esta query aprendemos a usar los *\$gte* y *\$lte* que son operadores de rango. En este caso rango de año entre 2012-2014 y que tengan entre 5 y 9 premios. Para la Shell encapsulamos la query dentro del find.



```
> use CinemApp
< switched to db CinemApp
> db.movies.find({ countries: "USA", year: { $gte: 2012, $lte: 2014 }, "awards.wins": { $gte: 5, $lte: 9 } })
{
  {
    _id: ObjectId('573a13acf29313caabd29366'),
    fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for the magazine and has a tedious life, not go
    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 journey that turns into an adventure =
    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'
    ],
    writers: [
      'Steve Conrad (screenplay)',
      'Steve Conrad (screen story by)',
      'James Thurber (based on the short story by)'
    ]
  }
}
```

En esta consulta se selecciona la colección *movies* de la base de datos *CinemApp*. Luego, se filtra el resultado

Nivell 2

- Exercici 1

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

22.841 de 50304 comentarios tienen el dominio *gameofthron.es*, el resto parecen usar *fakegmail.es*.

Para este ejercicio he usado la expresión regular `/\`, que sirve para filtrar un patrón de texto concreto, el punto está fuera de las cadenas porque es un punto literal, el `$` marca el final de la cadena y la `i` hace que sea sensible a mayúsculas o minúsculas.

```
> use CinemApp
< switched to db CinemApp
> db.comments.countDocuments({ email: /@gameofthron\.es$/i})
< 22841
CinemApp>
```

- Exercici 2

Quants cinemes hi ha en cada codi postal situats dins de l'estat Washington D. C. (DC)?

Para este ejercicio vamos a usar el método `aggregate()`, que es como una cadena de montaje de una fábrica dónde en cada etapa se procesan, filtran y agrupan los datos para que salgan como los necesitas. En este caso usando *match*, *group* y *sort* para localizar el estado concreto de *DC*, agrupando por *zipcode* y contando cuántos hay y luego *sort* para ordenarlos de mayor a menor para que salgan en los que hay más teatros primero..

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

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.

Este es muy parecido a uno que hemos realizado en el nivel anterior, ya que también usamos \$gte y \$lte para buscar una película con una puntuación mayor o igual a 7.5 y menor o igual a 8. usamos dos filtros, el de directores y el de imdb.rating.

```
{ "directors": "John Landis", "imdb.rating": { "$gte": 7.5, "$lte": 8 } }
```

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1-4 of 4

```
{
  "_id": ObjectId('573a1397f29313caabce6d94'),
  "fullplot": "Faber College has one frat house so disreputable it will take anyone. ...",
  "imdb": Object
    year: 1978
    plot: "At a 1962 college, Dean Vernon Wormer is determined to expel the entire..."
    genres: Array (1)
    rated: "R"
    metacritic: 82
    title: "Animal House"
    lastupdated: "2015-09-13 00:02:47.803000000"
    languages: Array (2)
    writers: Array (3)
    type: "movie"
    tomatoes: Object
      poster: "https://m.media-amazon.com/images/M/MV5BM2ZDA4NTYtOGJhMjI0OTg5LW1ZT..."
      num_mflix_comments: 1
      released: 1978-07-28T00:00:00.000+00:00
    awards: Object
    countries: Array (1)
    cast: Array (4)
    directors: Array (1)
    runtime: 109
  }
}
```

```
{
  "_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 are so sanctimonious...",
  "imdb": Object
    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 for him.",
    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/'
    }
  }
}
```

- Exercici 2

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

En este caso vamos a usar sólo la gui. Vamos a *theaters* y en este caso no vamos a *Documents* sino a la pestaña *Schema*. Usamos el botón *Analyze* y seleccionamos *geo*, aunque también nos serviría *coordinates*.

Documents1.6KAggregationsSchemaIndexes1Validation

Generate query +ResetAnalyzeOptions

Project{ field: 0 }

Sort{ field: -1 } or [['field', -1]]

Collation{ locale: 'simple' }

Index Hint{ field: -1 }

Max Time MS60000

skip0Limit0

The schema content is outdated and no longer in sync with the documents view. Press "Analyze" again to see the schema for the current query.

addressdocument

geocoordinates

Document with 5 nested fields.

