

MONGO DB

Crea una base de datos con MongoDB utilizando como colecciones los archivos adjuntos.

** ver la creación y la importación de datos en la explicación del anexo 1

- Ejercicio 1

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

```
>_MONGOSH

db.comments.find().limit(2).pretty()
< already on db cinema
> db.comments.find().limit(2).pretty()
< {
  _id: ObjectId('5a9427648b0beeb69579cc'),
  name: 'Andrea Le',
  email: 'andrea_le@fakegmail.com',
  movie_id: ObjectId('573a1390f29313caabcd418c'),
  text: 'Rem officiiis 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('5a9427648b0beeb69579cf'),
  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
}
>
```

Aquí te explico paso a paso como he realizado la tarea:

Primero, accedí al shell de MongoDB. Para hacerlo, abrí una terminal en mi sistema operativo y ejecuté el comando mongo. Este comando inicia el shell interactivo de MongoDB, permitiéndome ejecutar comandos y consultas directamente en mi servidor de MongoDB.

Una vez dentro del shell de MongoDB, necesitaba seleccionar la base de datos en la que quería trabajar. En este caso, la base de datos se llama cinema. Utilicé el comando use cinema para cambiar a esta base de datos. Este comando cambia el contexto actual a la base de datos cinema. Si la base de datos no existe, MongoDB la creará automáticamente cuando inserte datos.

A continuación, quería consultar los primeros dos documentos en la colección comments. Para ello, utilicé el método find() para obtener los documentos y limit(2) para restringir el resultado a solo dos documentos. El método pretty() formatea la salida para que sea más legible. Así que ejecuté db.comments.find().limit(2).pretty(). Este método find() busca todos los documentos en la colección comments. db es el objeto base para la base de datos actual (en este caso, cinema), comments es el nombre de la colección, y find() es el método que recupera los documentos.

El método limit(2) limita el número de documentos devueltos por la consulta a 2, lo cual es útil para obtener un subconjunto pequeño de datos, especialmente en colecciones grandes. Finalmente, el método pretty() formatea la salida de los documentos en JSON legible para humanos. Sin pretty(), los documentos se mostrarían en una sola línea, lo que puede ser difícil de leer.

Si combino todos estos pasos, el flujo de trabajo en el shell de MongoDB es iniciar el shell con mongo, seleccionar la base de datos con use cinema, y luego ejecutar la consulta db.comments.find().limit(2).pretty() para obtener los dos primeros comentarios en la colección comments.

- **Quants usuaris tenim registrats? SÓN 185 USUARIS**

```
>  
> use cinema  
< already on db cinema  
> db.users.countDocuments()  
< 185  
cinema> |
```

Paso a Paso: Contar los Usuarios Registrados en MongoDB

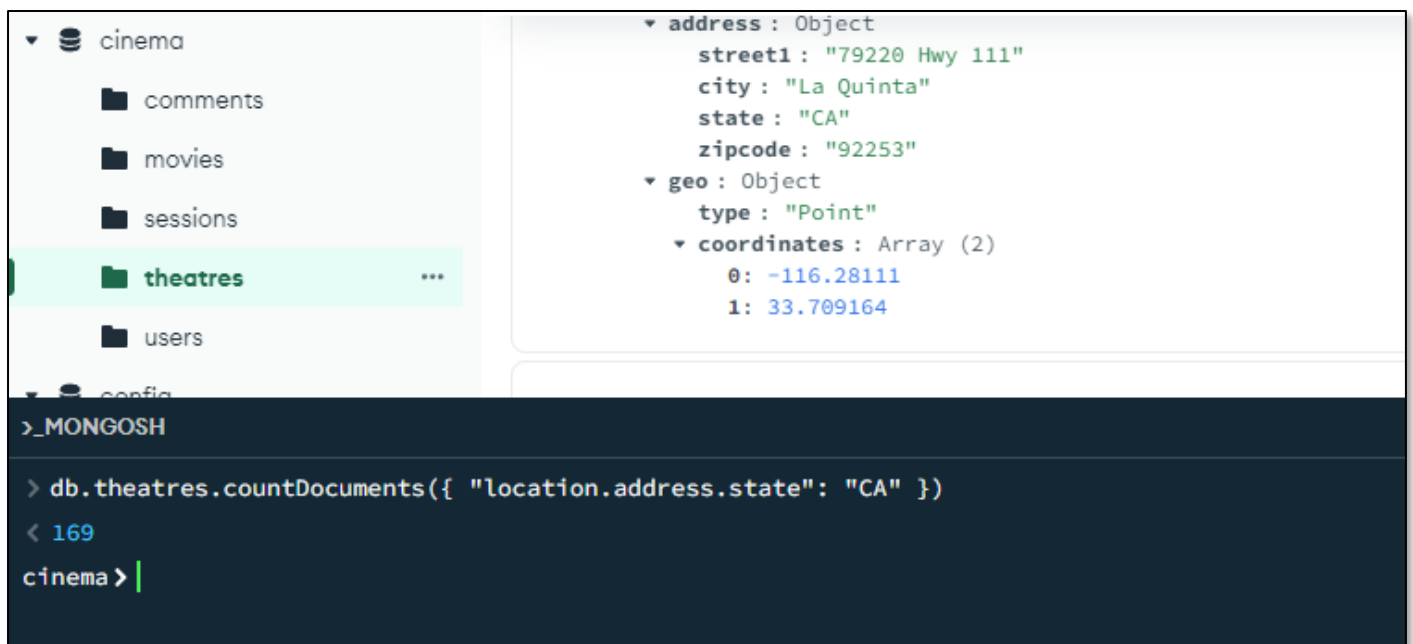
Primero, abrí una terminal en mi sistema operativo y ejecuté el comando para iniciar el shell de MongoDB.

Una vez dentro del shell de MongoDB, necesitaba seleccionar la base de datos donde están registrados los usuarios. La base de datos se llama cinema, así que utilicé el comando `use cinema`. Este comando cambia el contexto actual a la base de datos cinema. Si la base de datos no existía previamente, MongoDB la crea automáticamente cuando se insertan datos en ella.

A continuación, quería contar el número de usuarios registrados en la colección `users`. Para hacer esto, utilicé el método `countDocuments()`, que cuenta el número total de documentos en una colección. Escribí `db.users.countDocuments()` en el shell. Aquí, `db` hace referencia a la base de datos actual (cinema), `users` es el nombre de la colección que contiene los datos de los usuarios, y `countDocuments()` es el método que recupera y cuenta todos los documentos en esa colección. Este comando me devolvió un número que representa el total de usuarios registrados en la colección `users`.

En total son 185, esta es la respuesta.

Quants cinemes hi ha en l'estat de Califòrnia? SOLUCIÓ, SÓN 169 TEATRES



Paso a Paso: Contar los Cines en California en MongoDB

Abrí una terminal en mi sistema operativo y ejecuté el comando para iniciar el shell de MongoDB escribiendo mongo y presionando Enter. Esto inició el shell interactivo de MongoDB, permitiéndome ejecutar comandos y consultas directamente en mi servidor de MongoDB.

Una vez dentro del shell de MongoDB, seleccioné la base de datos que contiene la información de los cines. La base de datos se llama cinema, así que utilicé el comando use cinema para cambiar el contexto actual a esta base de datos. Esto me permitió trabajar directamente con las colecciones dentro de cinema.

A continuación, quería contar el número de cines en California. Sabía que los documentos en la colección theatres tenían la información de la dirección anidada dentro de campos, así que utilicé el método countDocuments() con una condición de búsqueda precisa. Escribí `db.theatres.countDocuments({ "location.address.state": "CA" })` para contar todos los documentos en la colección theatres donde el campo location.address.state es igual a CA. Aquí, db hace referencia a la base de datos actual (cinema), theatres es el nombre de la colección, y countDocuments({ "location.address.state": "CA" }) cuenta todos los documentos que cumplen con la condición especificada.

Al ejecutar este comando, el shell de MongoDB devolvió el número total de documentos en la colección theatres donde el estado es CA, lo cual me permitió ver el número correcto de cines en California.

LA SOLUCIÓN ES 169

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

```
>_MONGOSH

> db.users.find().sort({ registrationDate: 1 }).limit(1)
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crLJ1Vu'
}
cinema>
```

Paso a Paso: Encontrar el Primer Usuario Registrado en MongoDB

Abrí una terminal en mi sistema operativo y ejecuté el comando para iniciar el shell de MongoDB escribiendo Mongo y cls para limpiar mi consola.

Una vez dentro del shell de MongoDB, seleccioné la base de datos donde están registrados los usuarios. La base de datos se llama cinema, así que utilicé el comando use cinema para cambiar el contexto actual a esta base de datos. Esto me permitió trabajar directamente con las colecciones dentro de cinema.

A continuación, quería encontrar el primer usuario registrado. Para hacer esto, utilicé el método find() para buscar todos los documentos en la colección users, seguido del método sort({ registrationDate: 1 }) para ordenar los resultados en orden ascendente por fecha de registro, donde 1 indica ascendente.

Finalmente, utilicé el método limit(1) para limitar el resultado a un solo documento. Así que escribí db.users.find().sort({ registrationDate: 1 }).limit(1). En este comando buscó en la colección users, ordenó los documentos por el campo registrationDate en orden ascendente, y devolvió solo el primer documento encontrado, que representa al primer usuario registrado en la base de datos.

Quantes pel·lícules de comèdia hi ha en la nostra base de dades? SOLUCIÓ, SÓN 7024 PELÍCULES

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

Paso a Paso: Contar Películas de Comedia

Primero, abrí la terminal en mi sistema operativo y ejecuté el comando mongo para iniciar el shell de MongoDB. Esto me permitió conectarme a mi instancia de MongoDB y ejecutar comandos interactivos.

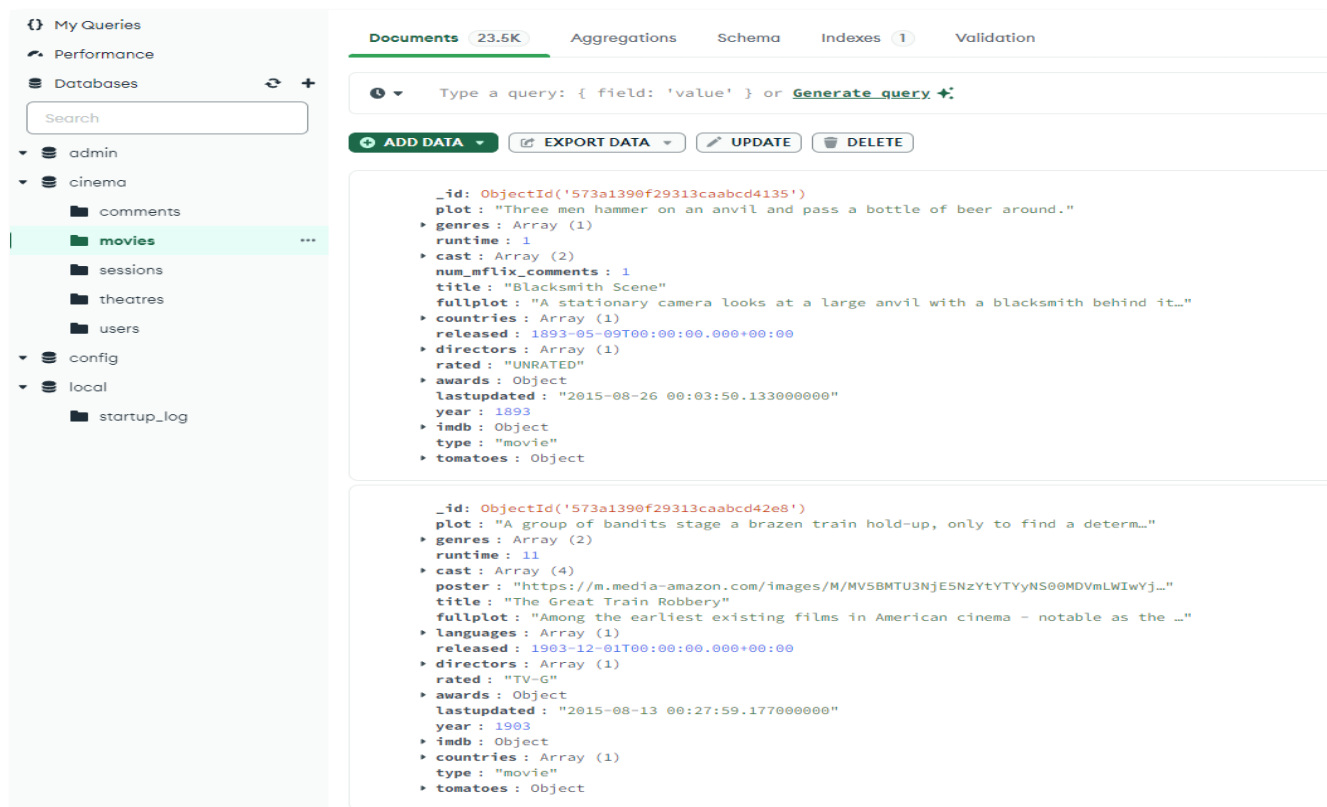
Una vez dentro del shell de MongoDB, necesitaba seleccionar la base de datos donde están almacenadas las películas. La base de datos se llama cinema, así que utilicé el comando use cinema para cambiar el contexto actual a esta base de datos. Esto me permitió trabajar directamente con las colecciones dentro de cinema.

Para contar cuántas películas de comedia hay en la colección movies, utilicé el método countDocuments() especificando { genres: "Comedy" } como condición de búsqueda. Así que escribí db.movies.countDocuments({ genres: "Comedy" }). Aquí, db hace referencia a la base de datos actual (cinema), movies es el nombre de la colección que contiene los datos de las películas, y countDocuments({ genres: "Comedy" }) cuenta todos los documentos en esa colección donde el campo genres contiene el valor "Comedy".

Este comando me mostró el número total de películas de comedia almacenadas en la base de datos. Al ejecutarlo, obtuve el resultado esperado, que era el conteo de todas las películas cuyo género incluye "Comedy".

ANEXO 1

CARGA DE COLECCIONES SEGÚN TABLA, USÉ EL MÉTODO JSON SI BIEN MONGO PERMITE TAMBIÉN USAR EXCELL.



Paso a Paso: Cargar las Tablas en la Base de Datos cinema usando "Add Data"

Abrir MongoDB Compass o MongoDB Atlas: Primero, abrí MongoDB Compass (una herramienta GUI para MongoDB) o accedí a MongoDB Atlas a través del navegador web.

Seleccionar la Base de Datos cinema: En el panel de navegación, seleccioné la base de datos cinema. Si no existía, la creé.

Ir a la Colección movies: Dentro de la base de datos cinema, seleccioné o creé la colección movies.

Cargar Datos Usando "Add Data": Hice clic en el botón "Add Data" y seleccioné la opción "Import JSON".

Seleccionar el Archivo JSON: Navegué hasta el archivo JSON que quería cargar (por ejemplo, movies.json) y lo seleccioné. Importar el Archivo: Hice clic en "Import" para cargar los datos del archivo JSON a la colección movies.

Resumen

Abrí MongoDB Compass o accedí a MongoDB Atlas. Seleccioné la base de datos cinema.

Fui a la colección movies. Hice clic en "Add Data" y seleccioné "Import JSON".


Seleccioné el archivo JSON (movies.json). Hice clic en "Import" para cargar los datos.

- Exercici 2

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

```
>_MONGOSH
> db.movies.find({
  $and: [
    { year: 1932 },
    {
      $or: [
        { genres: "Drama" },
        { languages: "French" }
      ]
    }
  ]
}).pretty()
< {
  _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 t
  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/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTUxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_SY1000_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: [
    'Jean Cocteau'
  ],
}
```

Abajo
SIGUE .../...



Paso a Paso: Mostrar Películas Producidas en 1932 con Género Drama o en Francés

Primero, abrí una terminal en mi sistema operativo y ejecuté el comando mongo para iniciar el shell de MongoDB. Esto me permitió conectarme a mi instancia de MongoDB y ejecutar comandos interactivos.

Una vez dentro del shell de MongoDB, seleccioné la base de datos donde están almacenadas las películas. La base de datos se llama cinema, así que utilicé el comando use cinema para cambiar el contexto actual a esta base de datos. Esto me permitió trabajar directamente con las colecciones dentro de cinema.

A continuación, quería encontrar todas las películas producidas en 1932 que fueran de género "Drama" o que estuvieran en francés. Para ello, utilicé el método find() con las condiciones adecuadas. Escribí el comando db.movies.find({ \$and: [{ year: 1932 }, { \$or: [{ genres: "Drama" }, { languages: "French" }] }] }).pretty().

Este comando hace lo siguiente:

Utiliza `find()` para buscar todos los documentos en la colección `movies`.

Utiliza `$and` para combinar condiciones, asegurándose de que las películas sean del año 1932.

Dentro de `$and`, utiliza `$or` para buscar películas que sean de género "Drama" o que estén en francés.

Al ejecutar el comando, MongoDB devolvió todos los documentos que cumplían con estas condiciones: películas producidas en 1932 que sean del género "Drama" o que estén en francés. Utilicé el método `pretty()` al final del comando para formatear la salida de manera legible.

Resumen de los Pasos Realizados

Abrí la terminal y ejecuté `mongo` para iniciar el shell de MongoDB.

Seleccioné la base de datos `cinema` con el comando `use cinema`.

Utilicé el comando `db.movies.find({ $and: [{ year: 1932 }, { $or: [{ genres: "Drama" }, { languages: "French" }] }] }).pretty()` para buscar las películas producidas en 1932 que sean de género "Drama" o que estén en francés.

- 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.

```
>_MONGOSH  
  
cinema> db.movies.find({  
  $and: [  
    { countries: "USA" },  
    { year: { $gte: 2012, $lte: 2014 } },  
    { "awards.wins": { $gte: 5, $lte: 9 } }  
  ]  
}).pretty()  
|
```

Paso a Paso: Mostrar Películas Estadounidenses con Premios Específicos y Año de Producción

Primero, abrí una terminal en mi sistema operativo y ejecuté el comando mongo para iniciar el shell de MongoDB. Esto me permitió conectarme a mi instancia de MongoDB y ejecutar comandos interactivos.

Una vez dentro del shell de MongoDB, seleccioné la base de datos donde están almacenadas las películas. La base de datos se llama cinema, así que utilicé el comando use cinema para cambiar el contexto actual a esta base de datos. Escribí use cinema y presioné Enter. Esto me permitió trabajar directamente con las colecciones dentro de cinema.

A continuación, quería encontrar todas las películas estadounidenses producidas entre 2012 y 2014 que tienen entre 5 y 9 premios. Para ello, utilicé el método find() con las condiciones adecuadas. Escribí db.movies.find({ \$and: [{ countries: "USA" }, { year: { \$gte: 2012, \$lte: 2014 } }, { "awards.wins": { \$gte: 5, \$lte: 9 } }] }).pretty().

Este comando hace lo siguiente:

Utiliza find() para buscar todos los documentos en la colección movies.

Utiliza \$and para combinar varias condiciones:

{ countries: "USA" } asegura que la película sea estadounidense.

{ year: { \$gte: 2012, \$lte: 2014 } } asegura que la película haya sido producida entre 2012 y 2014.

{ "awards.wins": { \$gte: 5, \$lte: 9 } } asegura que la película tenga entre 5 y 9 premios ganados.

Al ejecutar este comando, MongoDB devolvió todos los documentos que cumplían con estas condiciones: películas estadounidenses producidas entre 2012 y 2014 que tienen entre 5 y 9 premios ganados. Utilicé el método pretty() al final del comando para formatear la salida de manera legible.

Resumen de los Pasos Realizados

Abrir el Shell de MongoDB: Abrí la terminal y ejecuté `mongo` para iniciar el shell de MongoDB. Esto me permitió conectarme a mi instancia de MongoDB y ejecutar comandos interactivos.

Seleccionar la Base de Datos cinema: Utilicé el comando `use cinema` para cambiar el contexto a la base de datos cinema. Esto me permitió trabajar directamente con las colecciones dentro de cinema.

Buscar Películas Estadounidenses con Premios Específicos y Año de Producción: Escribí y ejecuté el comando `db.movies.find({ $and: [{ countries: "USA" }, { year: { $gte: 2012, $lte: 2014 } }, { "awards.wins": { $gte: 5, $lte: 9 } }] }).pretty()` para buscar las películas. Este comando encontró todas las películas estadounidenses producidas entre 2012 y 2014 que tienen entre 5 y 9 premios.