

Descripció

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

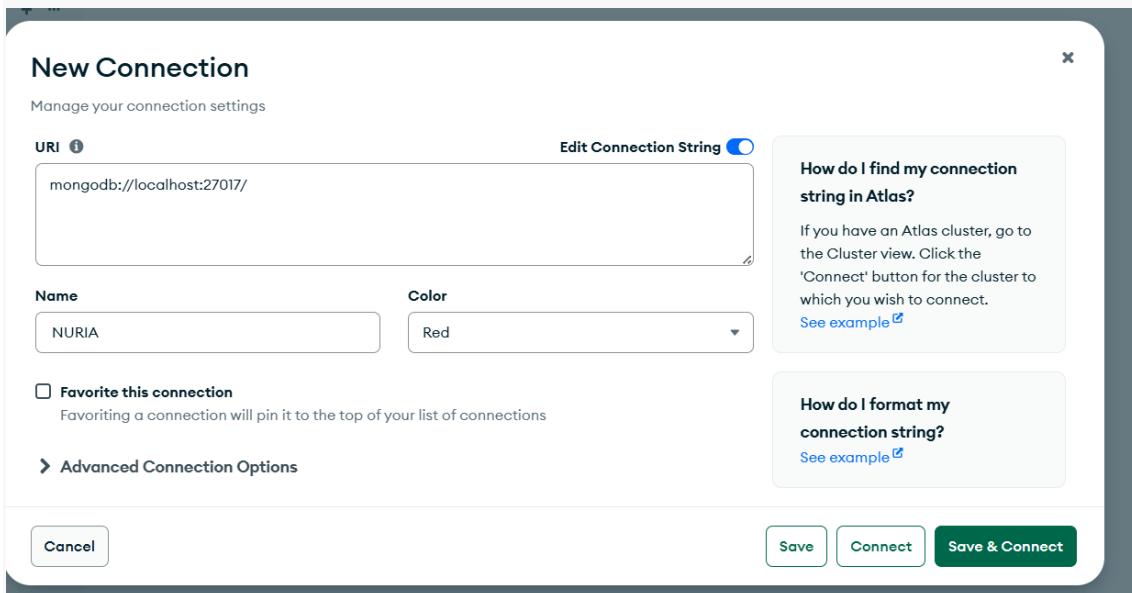
- users: Emmagatzema informació d'usuari/es, incloent-hi noms, emails i contrasenyes xifrades.
- theatres: Conté dades de cinemes, com ID, ubicació (direcció i coordenades geogràfiques).
- sessions: Guarda sessions d'usuari, incloent-hi ID d'usuari i tokens JWT per a l'autenticació.
- movies: Inclou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

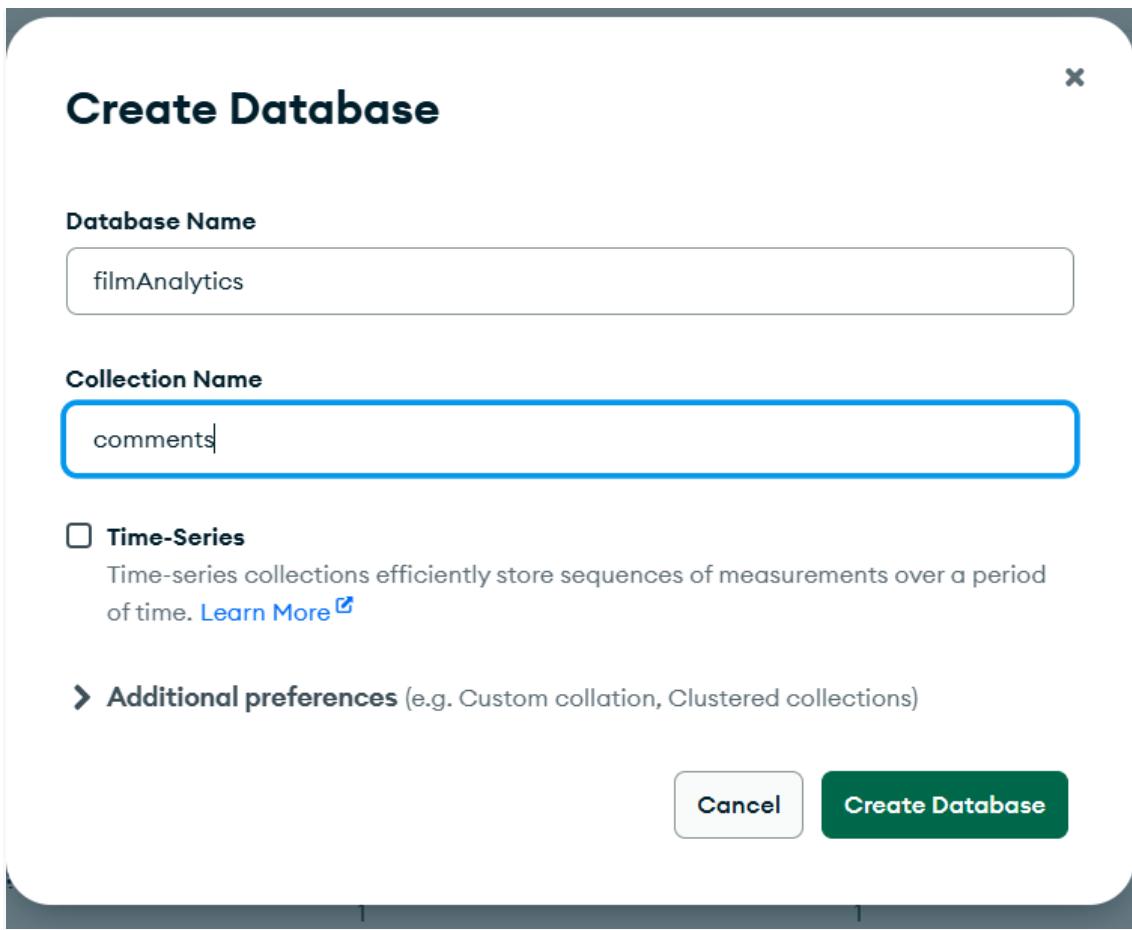
La pàgina de descàrrega de MongoDB Community Edition permet obtenir la versió gratuïta de la base de dades NoSQL. Descarreguem MongoDB

<https://www.mongodb.com/try/download/community>

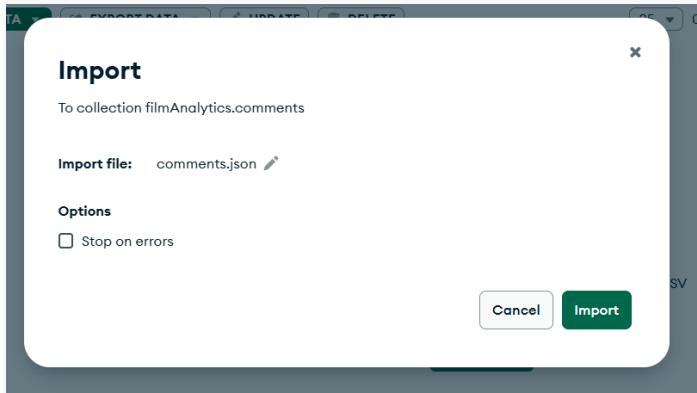
i creem la nostra BBDD: filmDB
podem crear una nova connexió o carregar la BBDD en el localhost:27017



creem la BBD i el nom de la primera col·lecció



i carreguem el primer doc json



Compass

localhost:27017 comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

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

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 50304

`_id: ObjectId('5a9427648b0beebe69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caab418c')
text: "Re officiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00`

`_id: ObjectId('5a9427648b0beebe69579cf')
name: "Greg Powell"
email: "greg_powell@fakegmail.com"
movie_id: ObjectId('573a1390f29313caab41b1')
text: "Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co..."
date: 1987-02-10T00:29:36.000+00:00`

`_id: ObjectId('5a9427648b0beebe69579d0')
name: "Talissa Maegyr"
email: "onaa_chaplin@gameofthron.es"
movie_id: ObjectId('573a1390f29313caab41b1')
text: "Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..."
date: 1998-08-22T11:45:03.000+00:00`

Import completed.
50304 documents imported.

i continuar amb la resta de documents amb **+** en la database **CREATE COLLECTION**

Compass

localhost:27017 comments

Documents 73.8K Aggregations Schema Indexes 1 Validation

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

[ADD DATA](#)

Create Collection

Collection Name: movies

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)

Cancel **Create Collection**

`_id: ObjectId('5a9427648b0beebe69579d3')
name: "Cameron Duran"
email: "cameron_duran@fakegmail.com"
movie_id: ObjectId('573a1390f29313caab4171')`

Import completed.
23539 documents imported.

ADD DATA

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Import data

Al final ens queda així:

Nivell 1

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

Import completed.
23539 documents imported.

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
comments	6.92 MB	50 K	284.00 B	1	524.29 kB
movies	21.55 MB	24 K	1.60 kB	1	364.54 kB
sessions	20.48 kB	1	540.00 B	1	20.48 kB
theaters	147.46 kB	1.6 K	223.00 B	1	32.77 kB
users	36.86 kB	185	159.00 B	1	20.48 kB

Les col·leccions, que són equivalents a les taules d'una base de dades relacional, són la forma en què Mongo guarda les dades. Cada col·lecció està formada per diversos documents, que són anàlegs a les files o registres de les bases de dades tradicionals.

Exercici 1

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

```
>_MONGOSH
> use filmDB
< switched to db filmDB
> db.comments.find({}, { text: 1, _id: 0 }).limit(2)
< [
  {
    text: 'Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde'
  }
  {
    text: 'Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Commodi nisi sit placeat rerum vero cupiditate neque. D'
  }
]
filmDB>
```

- db.comments: db → és la base de dades que tinc activa (després de fer use filmDB i comments → és la col·lecció on estan els documents (els comentaris).
- find({}, { text: 1, _id: 0 })

- find() → busca documents dins de la col·lecció.
- {} → primer paràmetre: filtre.
- {} buit significa no filtrar res, agafa tots els documents.
- { text: 1, _id: 0 } → segon paràmetre: projecció (quins camps mostrar).
- text: 1 → mostra el camp text.
- _id: 0 → no mostra l'_id (MongoDB sempre el retorna per defecte si no ho indiques).
- limit(2) Limita el resultat a només 2 documents. Així només veuré els 2 primers comentaris que MongoDB troba en la col·lecció.

- Quants usuaris tenim registrats?

185 usuaris

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

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

169 cinemes a Califòrnia

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

Ned Stark

```
>_MONGOSH
> db.users.countDocuments()
< 185
> db.users.find().sort({ createdAt: 1 }).limit(1)
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIj9RJAqMUQ74crlJ1Vu'
}
filmDB>
```

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

7024 pel·lícules de comèdia

NURIA > filmDB > movies

Documents 23.5K Aggregations Schema Indexes 1 Validation

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

year: 1917
imdb: Object
countries: Array (1)
type: "movie"
tomatoes: Object

id: ObjectId('573a1390f29313caabcd6377')

comprovació:

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

Exercici 2

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

18 pel·lícules

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

Generate Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

_id: ObjectId('573a1391f29313caabcd9458')
plot: "A young artist draws a face at a canvas on his easel. Suddenly the mou..."
runtime: 55
rated: "UNRATED"
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWETYjlmYi00NjA4LTk4ZW..."
title: "The Blood of a Poet"
lastupdated: "2015-09-16 13:13:05.537000000"
languages: Array (1)
released: 2010-05-20T00:00:00.000+00:00
directors: Array (1)
writers: Array (1)

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.

166 pel·lícules

```

{
  $untries: "USA",
  "awards.wins": {
    $gte: 5,
    $lte: 9
  },
  year: {
    $gte: 2012,
    $lte: 2014
  }
}
  
```

- Operador mongo entre: { \$gte: 5, \$lte: 9 }

Operador	Significat	Inclou el valor igual?
\$gt	més gran que (>)	✗ No
\$gte	més gran o igual que (≥)	✓ Sí
\$lt	més petit que (<)	✗ No
\$lte	més petit o igual que (≤)	✓ Sí

Nivell 2

Exercici 1

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

22.841 comentaris

```

>_MONGOSH
> db.comments.countDocuments({
  email: /@gameofthron\.es$/i
})
< 22841
filmDB>

```

- `countDocuments()` → compta el nombre de documents que compleixen la condició.
- `email` → és el camp on es desa el correu.
- `/@gameofthron\.es$/i` → és una expressió regular (regex) que busca tots els correus que:
 - acaben amb `@gameofthron.es`
 - el modificador `i` fa que no distingeixi majúscules/minúscules (`GAMEOFTHRON.ES`, `gameofthron.es`, etc.)

Exercici 2

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

3 cinemes, 1 de cada codi postal

- `$match` → filtra només els cinemes situats a l'estat *DC*
- `$group` → agrupa els documents per codi postal (`zipcode`)
 - `_id` → representa el codi postal
 - `total_cinemes` → compta quants cinemes hi ha a cada codi postal
- `$sort` → ordena els resultats per codi postal

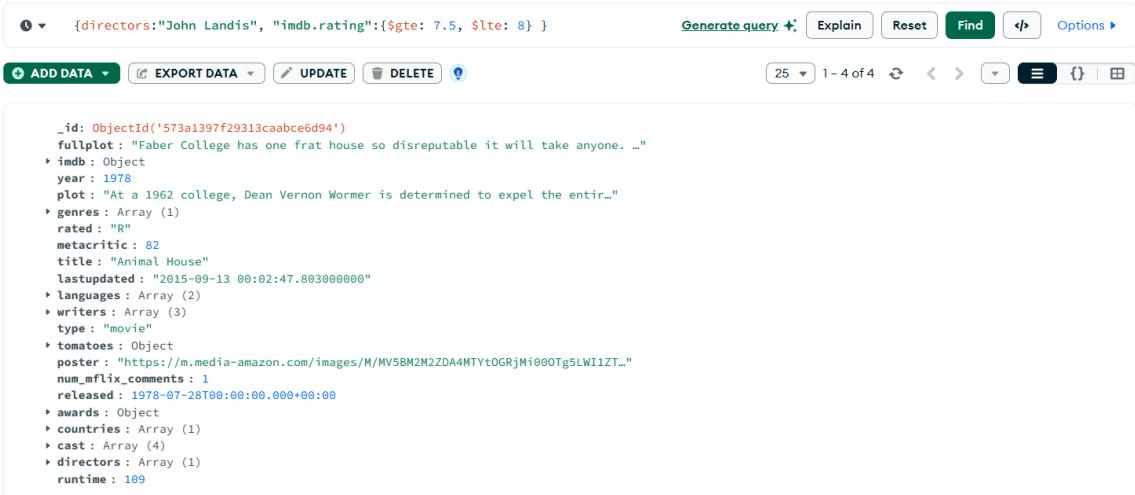
```
> db.theaters.aggregate([
  { $match: { "location.address.state": "DC" } },
  {
    $group: {
      _id: "$location.address.zipcode",
      total_cinemes: { $sum: 1 }
    }
  },
  { $sort: { _id: 1 } }
])
< [
  {
    _id: '20002',
    total_cinemes: 1
  },
  {
    _id: '20010',
    total_cinemes: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

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.

4 pel·lícules



```

_id: ObjectId('573a1397f29313caabce6d94')
fullplot: "Faber College has one frat house so disreputable it will take anyone. ..."
  _id: ObjectId('573a1397f29313caabce6d94')
  fullplot: "Faber College has one frat house so disreputable it will take anyone. ..."
  year: 1978
  plot: "At a 1962 college, Dean Vernon Wormer is determined to expel the entir..."
  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/MV5BM2M2ZDA4MTYtOGRjMi000Tg5LW1ZT..."
  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

```

Exercici 2

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

A schema obrim el mapa

