

Tasca S9.01. Consultes amb MongoDB

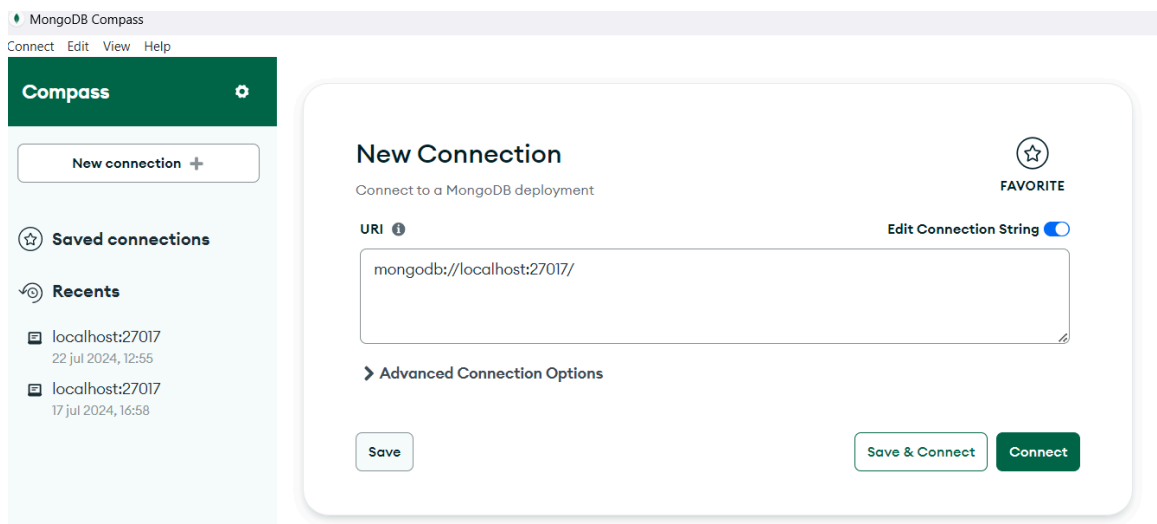
Treballarem amb una base de dades que conté col·leccions relacionades amb una aplicació d'entreteniment cinematogràfic:

- **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.

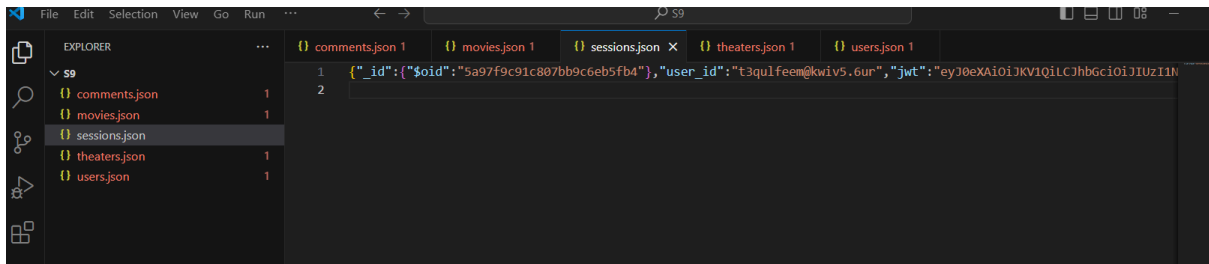
Nivell 1

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



Primer de tot, creem una base de dades amb el nom de **test** i triem una primera col·lecció, per exemple "comments". Una vegada creada la base de dades i importat el primer fitxer JSON, fem la importació de la resta d'arxius. Com que mai he treballat

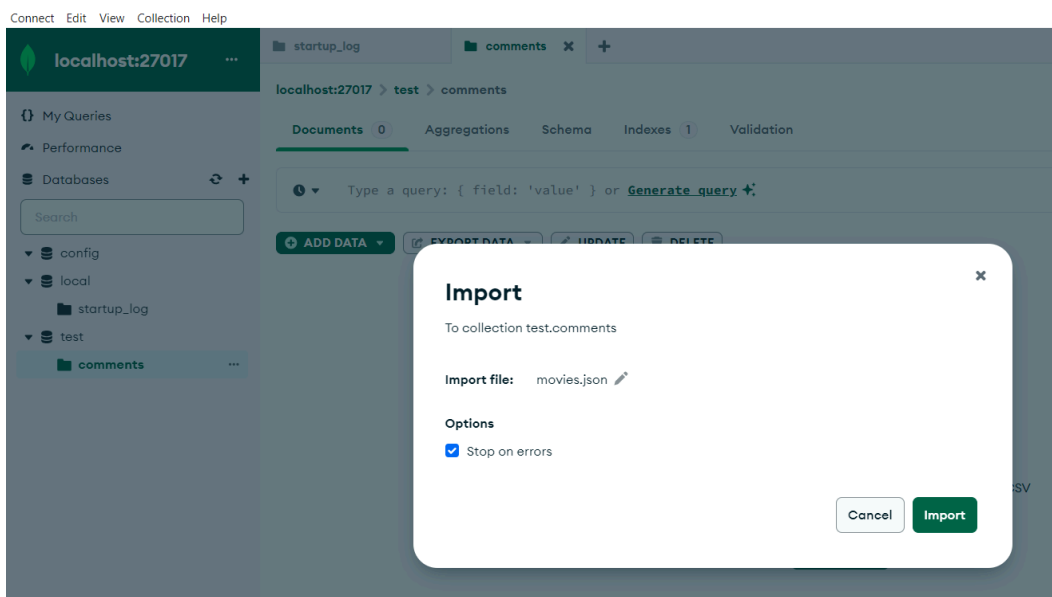
amb arxius JSON abans obro els arxius des del Visual Studio per explorar una mica les dades:



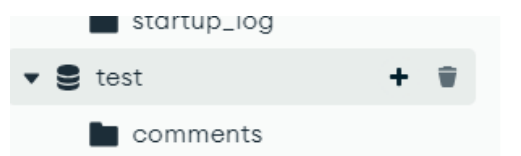
Observo que, com esperàvem, el format és vàlid per fer una importació a *MongoDB Compass*, ja que els arxius estan separats per comes en una matriu.

Així doncs, ja podem fer els passos per fer la importació de la resta d'arxius JSON en una col·lecció:

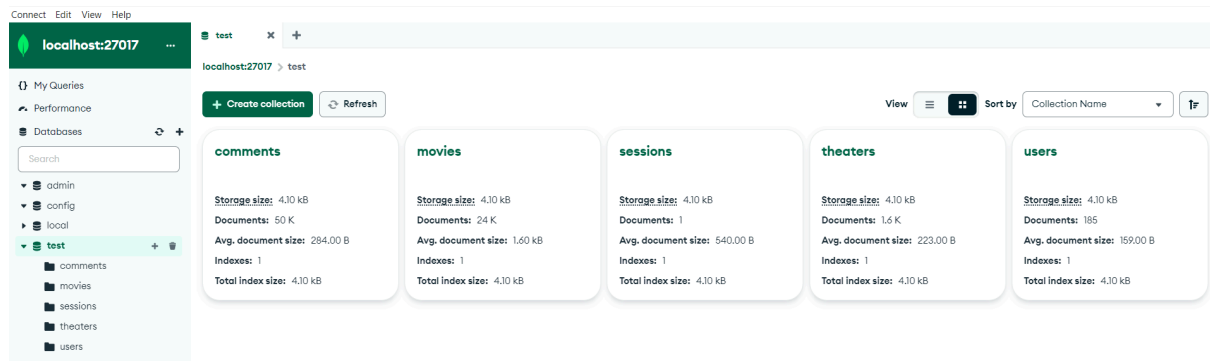
1. Seleccionada la base de dades **test** cliquem *ADD DATA* i seleccionem *Import JSON o CSV file*.
2. Importem "movies.json", marcant l'opció perquè es pari en cas d'error:



3. Ens informa que el procés s'ha fet correctament, per la qual cosa ara anem al símbol "+" que hi ha al costat de la base de dades **test** i creem la resta de col·leccions:



4. Hem carregat tots els arxius JSON en cinc col·leccions:

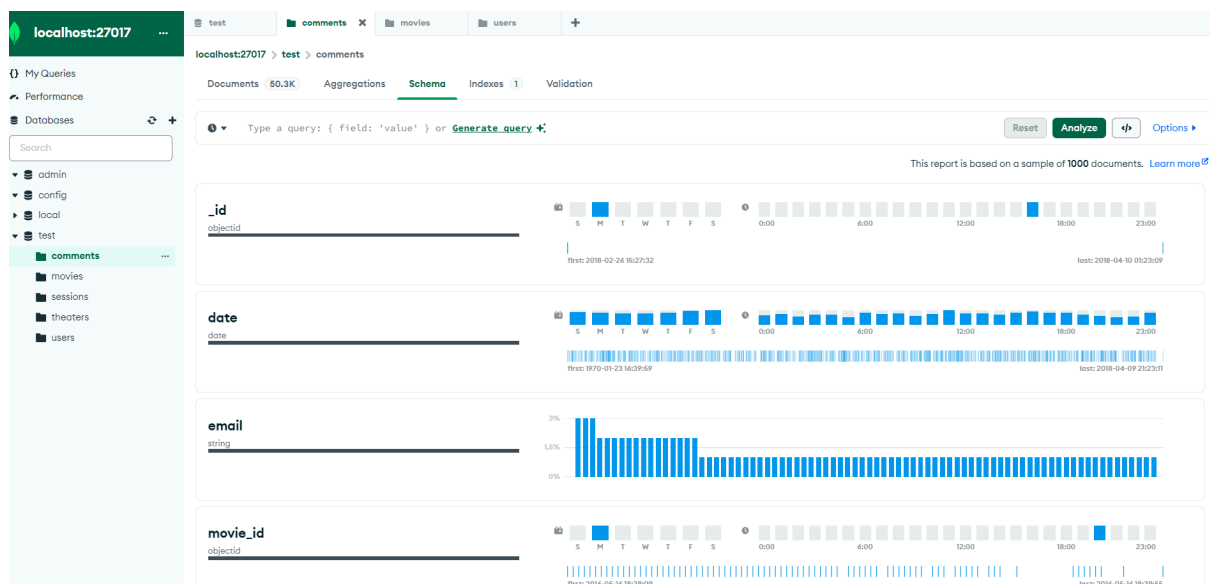


Test és la nostra bd, i les **col·leccions** són: **comments**, **movies**, **sessions**, **theaters** i **users**. Per fer un símil amb SQL, les col·leccions serien com l'equivalent a les taules.

Abans de passar a fer les consultes, explorem de manera ràpida l'esquema de cada col·lecció mitjançant l'opció d'anàlisi de l'esquema que ens ofereix informació sobre la freqüència, tipus i els intervals de camps del conjunt de dades.

Cal recordar que, gràcies a aquest modelatge flexible de dades de MongoDB, hem pogut crear les col·leccions a partir dels documents amb diferents estructures.

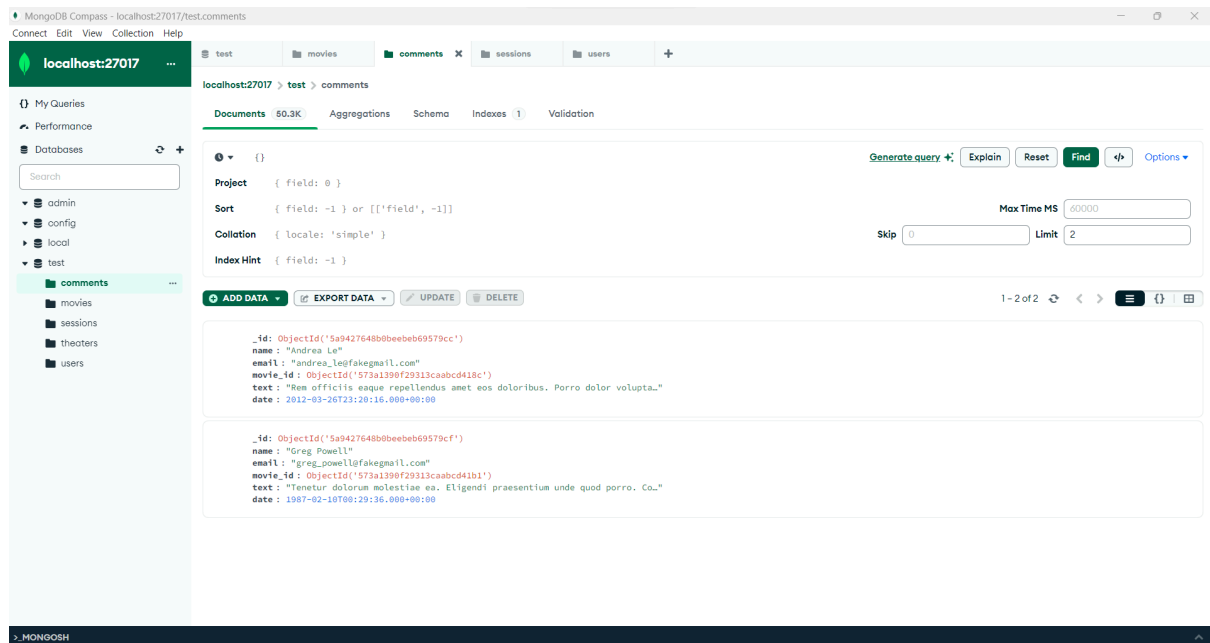
A continuació una mostra de l'esquema de "comments" que es genera a partir d'una mostra de 1.000 documents, que serien com els registres. Ens permet veure un resum de les dades d'una col·lecció.



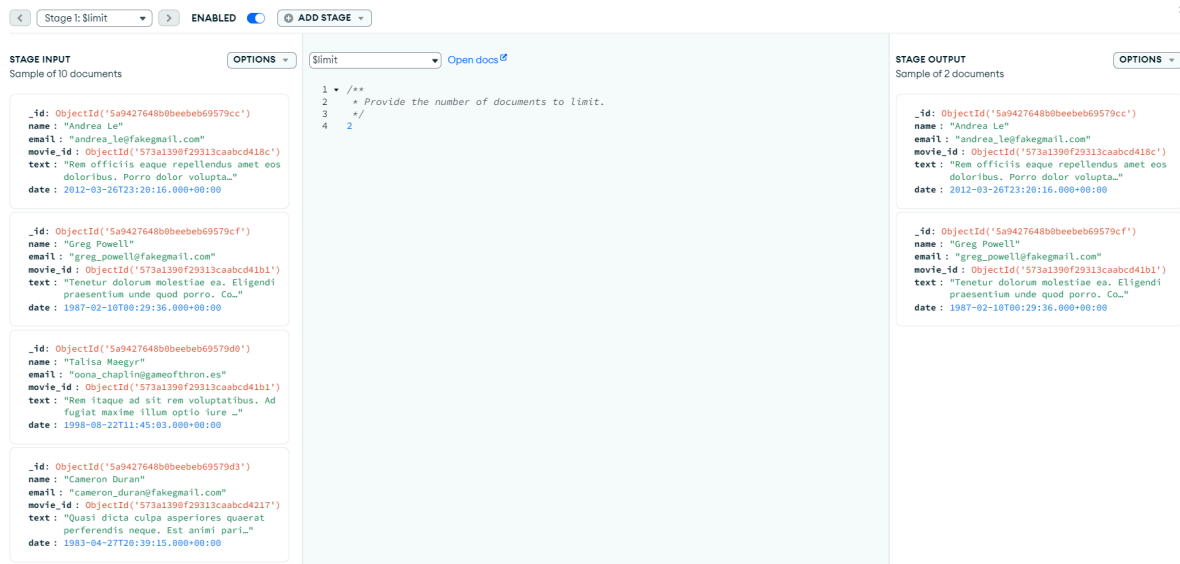
- Exercici 1

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

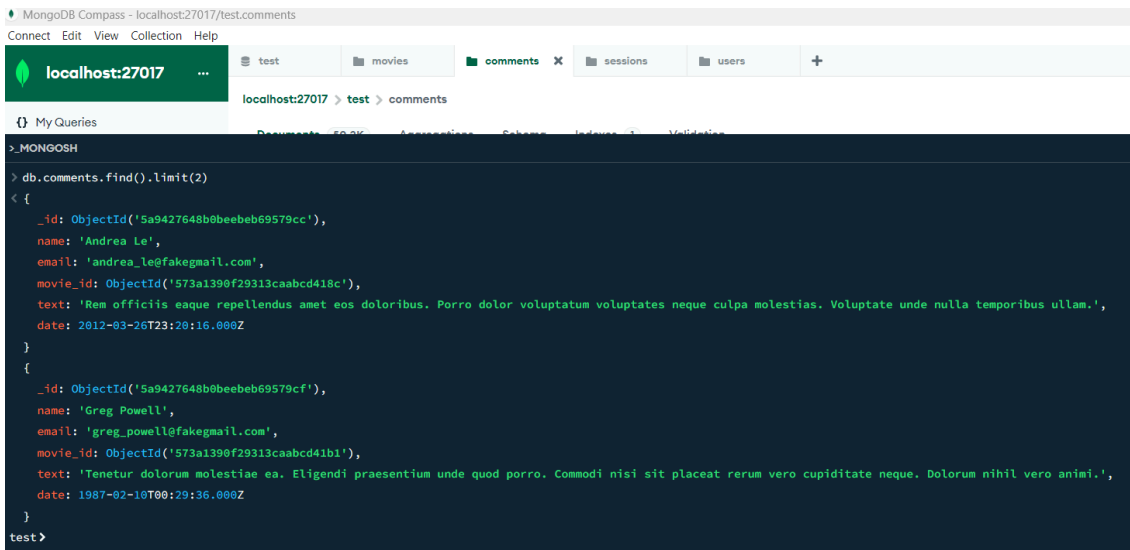
Una manera és utilitzant la interfície de *Compass*, des de la col·lecció “comments”, la pestanya documents, fer clic a les Opcions i limitar a 2:



També podem trobar els dos primers resultats amb Aggregations --> Add Stage --> \$limit --> posar el 2 a number:



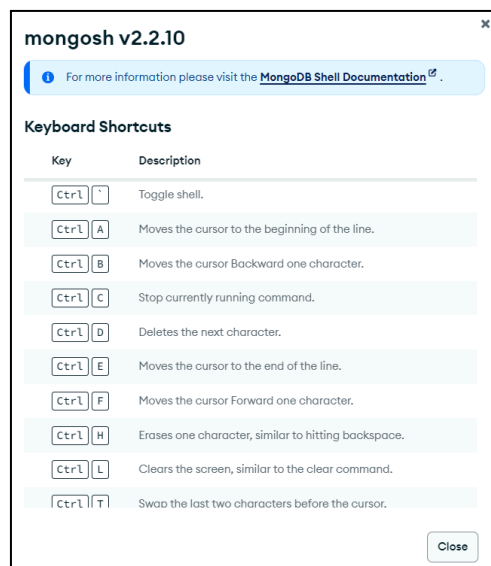
I en últim lloc podem obtenir els mateixos resultats utilitzant el shell incrustat `mongosh` de la següent manera:



The screenshot shows the MongoDB Compass interface. At the top, there's a header with 'MongoDB Compass - localhost:27017/test.comments'. Below it, a toolbar contains 'Connect', 'Edit', 'View', 'Collection', and 'Help'. A sidebar on the left shows the database structure: 'localhost:27017' with collections 'test', 'movies', 'comments', 'sessions', and 'users'. The main panel displays the MongoDB Shell with the command 'db.comments.find().limit(2)' and its output. The output shows two documents from the 'comments' collection.

```
>_MONGOSH
> db.comments.find().limit(2)
< {
  _id: ObjectId('5a9427648b0beeb69579cc'),
  name: 'Andrea Le',
  email: 'andrea_le@fakegmail.com',
  movie_id: ObjectId('573a1390f29313caabcd418c'),
  text: 'Rem officii 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
}
test>
```

Aquí ens faciliten els *shortcuts* per agilitzar el procés, i un enllaç per accedir a la Documentació:



The screenshot shows a dialog box titled 'mongosh v2.2.10'. It contains a link to the 'MongoDB Shell Documentation' and a table of keyboard shortcuts.

Key	Description
Ctrl `	Toggle shell.
Ctrl A	Moves the cursor to the beginning of the line.
Ctrl B	Moves the cursor Backward one character.
Ctrl C	Stop currently running command.
Ctrl D	Deletes the next character.
Ctrl E	Moves the cursor to the end of the line.
Ctrl F	Moves the cursor Forward one character.
Ctrl H	Erases one character, similar to hitting backspace.
Ctrl L	Clears the screen, similar to the clear command.
Ctrl T	Swaps the last two characters before the cursor.

Close

2. Quants usuaris tenim registrats?

Fent clic a *Aggregations* de la col·lecció *users* ja ens informa que hi ha 185 documents que equival a usuaris registrats, però repetim els mateixos passos des d'*Aggregations* --> Add Stage --> \$count :

localhost:27017 > test > users

Documents 185 Aggregations Schema Indexes 1 Validation

Count Generate aggregation + Explain Export Run Options

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

password : "\$2b\$12\$UREFwsRUoyF0CRqGNK0Lz00HM/j..." password : "\$2b\$12\$yGqxLG9LZpXA2xVDhuPnSOZd.VU..." password : "\$2b\$12\$6vz7wiwO.EtSRiLvq1zUc./9488..." password : "\$2b\$12\$fia1..."

Stage 1 Count

```

1 //**
2 * Provide the field name for the count.
3 */
4 'string'

```

Output after Count stage (Sample of 1 document)

```
string: 185
```

Amb **mongosh** obtenim el mateix resultat de **185 usuaris registrats**, utilitzant la funció “countDocuments()”:

MongoDB Compass - localhost:27017/test.users

Connect Edit View Collection Help

localhost:27017 test movies comments sessions users

localhost:27017 > test > users

Documents 185 Aggregations Schema Indexes 1 Validation

> MONGOSH

```

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

```

3. Quants cinemes hi ha en l'estat de Califòrnia?

Primer he anat a l'esquema i he vist que els documents tenen una estructura anidada. Aquesta informació la trobem de manera que el camp 'state' està anidat dins 'address' i finalment dins de 'location'. Califòrnia es representa com a CA i és de tipus cadena.

localhost:27017 > test > theaters

Documents 1.6K Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or Generate query + Reset Analyze Options

This report is based on a sample of 1000 documents. Learn more

location Document with 2 nested fields.

address Document with 5 nested fields.

city string

state string

Així doncs des de la interfície anem a *theaters* --> Stage 1 **\$match** i substituïm *query* per: **"location.address.state" : "CA"**

Per accedir a camps anidats utilitzem el **.** i com a filtre d'igualtat que busqui el valor **"CA"** perquè coincideixi en la nostra consulta. Amb això l'output ens retorna 10 documents, però encara ens falta un últim pas, que és fer el recompte de cinemes amb un Stage 2 **\$count**. El resultat ara sí, és de 169:

The screenshot shows the MongoDB Compass interface for the 'test' database, 'theaters' collection. The 'Aggregations' tab is active, showing two stages:

- Stage 1: \$match**
 - Query:

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "location.address.state" : "CA"
6 }
```
 - Output: Sample of 10 documents showing theater IDs and locations.
- Stage 2: \$count**
 - Field: `'string'`
 - Output: `string : 169`

Ara utilitzant **mongosh** també ens mostra que Califòrnia té 169 cinemes:

The screenshot shows the MongoDB Compass interface with the 'mongosh' terminal open. The terminal output is as follows:

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

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

Des de *Compass* col·lecció "users", pestanya de documents, fer clic a les Opcions i limitar a 1:

users x +

localhost:27017 > test > users

Documents 185 Aggregations Schema Indexes 1 Validation

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

Project { field: 0 }

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

Collation { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 60000

Skip 0 Limit 1

ADD DATA EXPORT DATA UPDATE DELETE

1 - 1 of 1

```

_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNI39RJAqMUQ74cr1J1Vu"

```

També així:

users x +

localhost:27017 > test > users

Documents 185 Aggregations Schema Indexes 1 Validation

Generate aggregation

Limit

PREVIEW STAGES TEXT WIZARD

password: "\$2b\$12\$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNI39RJAqMUQ74cr1J1Vu"

Stage 1 \$limit

1 // **
2 * Provide the number of documents to limit
3 */
4 1

Output after \$limit stage (Sample of 1 document)

```

_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNI39RJAqMUQ74cr1J1Vu"

```

+ Add Stage

I per acabar amb `mongo`, el resultat sempre és l'usuari amb el nom **Ned Stark**:

```

>_MONGO
> db.users.findOne()
< {
  _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
  name: 'Ned Stark',
  email: 'sean_bean@gameofthron.es',
  password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNI39RJAqMUQ74cr1J1Vu'
}
test>

```


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

Anem a la col·lecció “movies” des de l'esquema veiem que type és un *string* amb dos valors: “movie” i “series”:



Amb la consola és més ràpid:

```
> _MONGOSH
> db.movies.distinct("type")
< [ 'movie', 'series' ]
test>
```

Abans també vull veure la classificació de gèneres existent, torno a utilitzar la consola per veure com s'anomenen els gèneres:

```
> _MONGOSH
> db.movies.distinct("genres")
< [
  'Action',      'Adventure', 'Animation',
  'Biography',   'Comedy',   'Crime',
  'Documentary', 'Drama',   'Family',
  'Fantasy',     'Film-Noir', 'History',
  'Horror',      'Music',   'Musical',
  'Mystery',     'News',    'Romance',
  'Sci-Fi',      'Short',   'Sport',
  'Talk-Show',   'Thriller', 'War',
  'Western'
]
test>
```

Veiem, doncs, que hem de comptar el total de pel·lícules on gènere és **“Comedy”**.

Una manera d'obtenir el resultat és des d'*Aggregations* amb aquestes tres fases:

▼ Stage 1 \$match

```

1 1 /**
2 2 * query: The query in MQL.
3 3 */
4 4 {
5 5   type: "movie"
6 6 }

```

Output after \$match stage (Sample of 10 documents)

```

released: 1893-05-09T00:00:00.000+00:00
directors: Array (1)
rated: "UNRATED"
awards: Object
lastupdated: "2015-08-26 00:03:50.133000000"
year: 1893
imdb: Object
type: "movie"
tomatoes: Object

```

▼ Stage 2 \$match

```

1 1 /**
2 2 * query: The query in MQL.
3 3 */
4 4 {
5 5   genres: "Comedy"
6 6 }

```

Output after \$match stage (Sample of 10 documents)

```

cast: Array (1)
0: "Winsor McCay"
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYzg2NjNhNTctM...
title: "Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving ...
fullplot: "Cartoonist Winsor McCay agrees to create a large set of drawings that ...
poster: "https://m.media-amazon.com/images/M/MV5BMTQxNzI4ODQ3N...
title: "Gertie the Dinosaur"
fullplot: "Winsor Z. McCay bets another cartoonist that he can animate a dinosaur..."
languages: Array (1)
0: "English"
released: 1914-09-15T00:00:00.000+00:00
directors: Array (1)

```

▼ Stage 3 \$count

```

1 1 /**
2 2 * Provide the field name for the count.
3 3 */
4 4 "string"

```

Output after \$count stage (Sample of 1 document)

```

string: 7002

```

El resultat són **7002** pel·lícules de comèdia a la base de dades.

També ho podem obtenir així:

My Queries

>_MONGOSH

```

> db.movies.countDocuments({"$and": [ {"type": "movie"}, {"genres": "Comedy"}]})
< 7002
test>

```

- Exercici 2

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

Ens demanen tots els documents de la col·lecció "movies" que compleixin aquestes dues condicions:

1. Les pel·lícules han de ser produïdes en 1932 (ho trobem a {"type": "movie"}, {"year": 1932}).
2. El gènere ha de ser drama (ho trobem a {"genres": "Drama"}) o l'idioma ha de ser francès (ho trobem a {"languages": "French"}).

Consideració: Tot i veure que a `db.movies.distinct("languages")` ens apareix també el francès amb llengua dels signes, en aquest cas no l'inclourem.

```
'Finnish',
'Flemish',
'French',
'French Sign Language',
```

Obtenim els resultats amb *Compass*:

Stage 1 \$match

```
{type: "movie"}
```

Stage 1 \$match

```
1 // **
2 // query: The query in MQL.
3 // **
4 {
5   type: "movie"
6 }
```

Output after \$match stage (Sample of 10 documents)

```
{ "_id": "ObjectId('573a1390f29313caabed4135')",
  "plot": "Three men hammer on an anvil and pass a bottle of beer around.",
  "genres": Array (1),
  "runtime": 1,
  "num_mflix_comments": 1,
  "title": "Blacksmith Scene",
  "fullplot": "A stationary camera looks at a" },
{ "_id": "ObjectId('573a1390f29313caabed42e8')",
  "plot": "A group of bandits stage a brazen train hold-up, only to find a determ...",
  "genres": Array (2),
  "runtime": 11,
  "cast": Array (4),
  "poster": "https://m.media-amazon.com/images/M/MV5BMTU3NjESNzYTY...",
  "title": "The Great Train Robbery" },
{ "_id": "ObjectId('573a1390f29313caabed4323')",
  "plot": "A young boy, oppressed by his mother, goes on an outing in the country ...",
  "genres": Array (3),
  "runtime": 14,
  "rated": "UNRATED",
  "cast": Array (4),
  "num_mflix_comments": 2,
  "poster": "https://m.media-..." }
```

Stage 2 \$match

```
{year: 1932}
```

Stage 2 \$match

```
1 // **
2 // query: The query in MQL.
3 // **
4 {
5   year: 1932
6 }
```

Output after \$match stage (Sample of 10 documents)

```
{ "_id": "ObjectId('573a1391f29313caabed9458')",
  "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/MV5BYWY3ODk5ZWYt...",
  "title": "The Blue Light" },
{ "_id": "ObjectId('573a1392f29313caabed99a3')",
  "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/MV5BNTQ1NTNzMTQ1O...",
  "title": "The Blue Light" },
{ "_id": "ObjectId('573a1392f29313caabed99a3')",
  "plot": "A young French soldier in World War : is overcome with guilt when he k...",
  "genres": Array (1),
  "runtime": 76,
  "cast": Array (4),
  "num_mflix_comments": 2,
  "poster": "https://m.media-amazon.com/images/M/MV5BYzJlMDZjM2Q...",
  "title": "The Blue Light" }
```

Stage 3 \$match

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

Stage 3 \$match

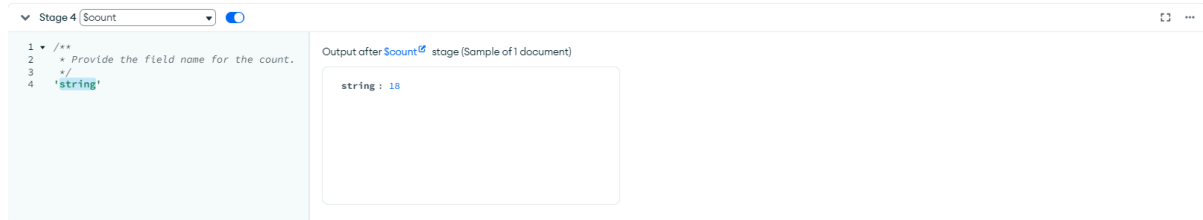
```
1 // **
2 // query: The query in MQL.
3 // **
4 {
5   $or: [
6     { genres: "Drama" },
7     { languages: "French" }
8   ]
9 }
```

Output after \$match stage (Sample of 10 documents)

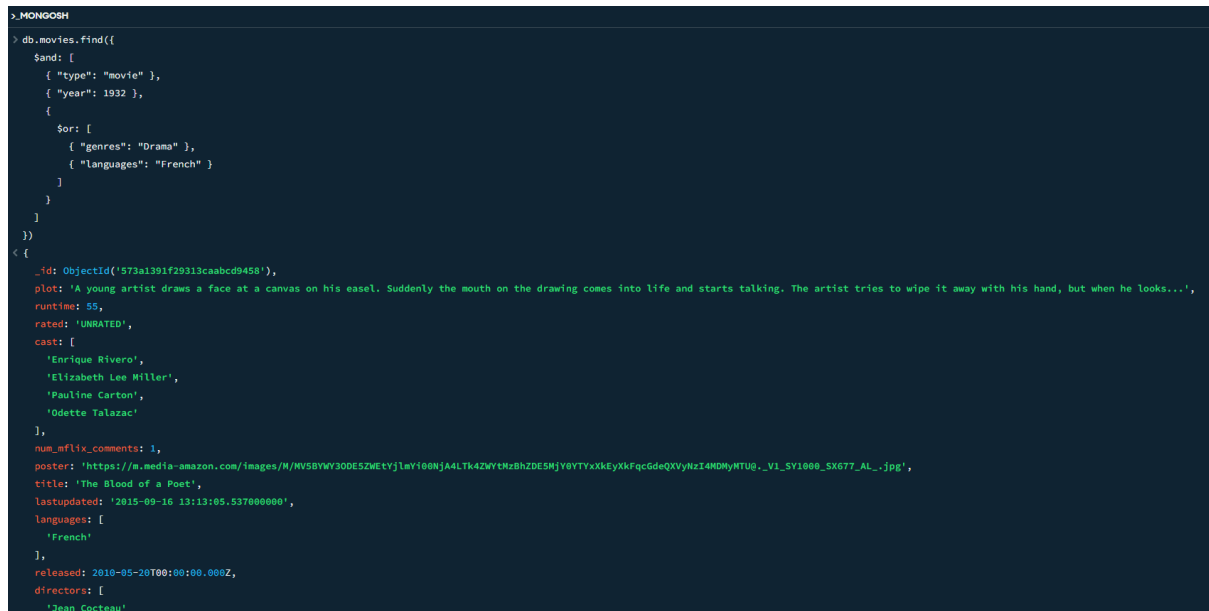
```
{ "_id": "ObjectId('573a1391f29313caabed9458')",
  "plot": "A young artist draws a face at a canvas on his easel. Suddenly the mou...",
  "runtime": 55,
  "rated": "UNRATED",
  "cast": Array (4),
  "num_mflix_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/MV5BYWY3ODk5ZWYt...",
  "title": "The Blue Light" },
{ "_id": "ObjectId('573a1392f29313caabed99a3')",
  "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/MV5BNTQ1NTNzMTQ1O...",
  "title": "The Blue Light" },
{ "_id": "ObjectId('573a1392f29313caabed99a3')",
  "plot": "A young French soldier in World War : is overcome with guilt when he k...",
  "genres": Array (1),
  "runtime": 76,
  "cast": Array (4),
  "num_mflix_comments": 2,
  "poster": "https://m.media-amazon.com/images/M/MV5BYzJlMDZjM2Q...",
  "title": "The Blue Light" }
```

Stage 4 \$count

De la consulta n'extreiem que tenim 18 documents amb aquestes condicions.



Amb la consola tots els resultats es veuen aquí:



I si fem un **countDocuments** obtenim un total de **18** documents.



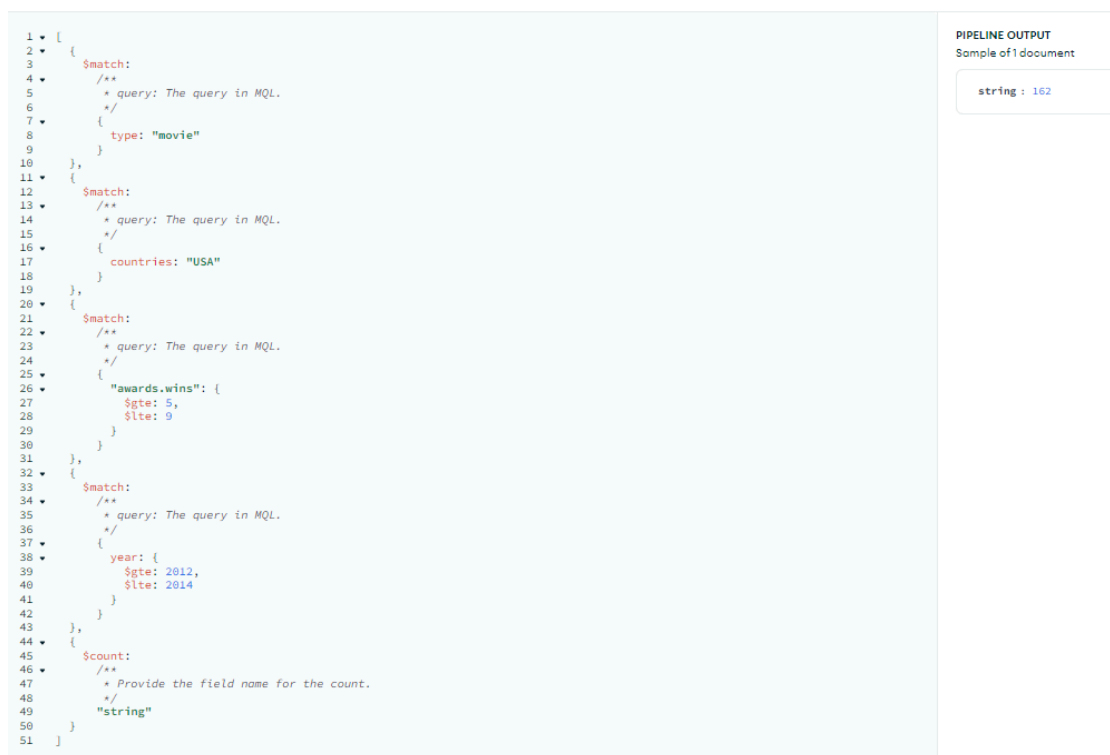
- Exercici 3

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

Per aquest cas ens demanen tots els documents de la col·lecció “movies” que compleixin aquests requisits:

1. Pel·lícules estatunidenques: El camp “type” ha de ser “movie” i “countries” ha de contenir el valor “USA”.
2. Entre 5 i 9 premis: El camp “awards.wins” ha de contenir un nombre de premis entre 5 i 9.
3. Produïdes entre 2012 i 2014: El camp “year” ha de ser entre 2012 i 2014.

Amb *Compass* des de la visualització </> text l'output final ens dona **162** documents:



The screenshot shows the MongoDB Compass interface. On the left, a query is written in the JSON view:

```
1 {
2   {
3     $match:
4     /**
5      * query: The query in MQL.
6      */
7     {
8       type: "movie"
9     }
10  },
11  {
12    $match:
13    /**
14     * query: The query in MQL.
15     */
16    {
17      countries: "USA"
18    }
19  },
20  {
21    $match:
22    /**
23     * query: The query in MQL.
24     */
25    {
26      "awards.wins": {
27        $gte: 5,
28        $lte: 9
29      }
30    }
31  },
32  {
33    $match:
34    /**
35     * query: The query in MQL.
36     */
37    {
38      year: {
39        $gte: 2012,
40        $lte: 2014
41      }
42    }
43  },
44  {
45    $count:
46    /**
47     * Provide the field name for the count.
48     */
49    "string"
50  }
51 }
```

On the right, the 'PIPELINE OUTPUT' section shows a sample of 1 document:

```
string : 162
```

Amb la consola fem:

```
>_MONGOSH
> db.movies.countDocuments({
  "type": "movie",
  "countries": "USA",
  "awards.wins": { $gte: 5, $lte: 9 },
  "year": { $gte: 2012, $lte: 2014 }
})
< 162
test>
```

Alguns títols d'aquestes pel·lícules són: 'The Secret Life of Walter Mitty', 'The Croods' o 'The Book Thief' com es pot veure aquí:

```
> db.movies.find(
  {
    "type": "movie",
    "countries": "USA",
    "awards.wins": { $gte: 5, $lte: 9 },
    "year": { $gte: 2012, $lte: 2014 }
  },
  {
    title: 1
  }
).limit(15)
< {
  _id: ObjectId('573a13acf29313caabd29366'),
  title: 'The Secret Life of Walter Mitty'
}
{
  _id: ObjectId('573a13b5f29313caabd45772'),
  title: 'The Croods'
}
{
  _id: ObjectId('573a13b9f29313caabd4dd15'),
  title: 'The Book Thief'
}
{
  _id: ObjectId('573a13b9f29313caabd4de39'),
  title: 'World War Z'
}
{
  _id: ObjectId('573a13b9f29313caabd4f21f'),
  title: 'Godzilla'
}
{
  _id: ObjectId('573a13bbf29313caabd5271d'),
  title: 'Mr Hockey: The Gordie Howe Story'
}
{
  _id: ObjectId('573a13bcf29313caabd57341'),
  title: 'Hitchcock'
}
```

Nivell 2

- Exercici 1

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

Fem la consulta directament a la consola, l'operador **\$regex** permet buscar els documents basats en patrons de text mitjançant expressions regulars. Permet cercar coincidències en camps de text, en el nostre cas escrivint **@gameofthron.es\$** tenim en compte que el correu ha d'acabar d'aquesta manera.

```
>_MONGOSH
> db.comments.countDocuments({
  "email": { $regex: /@gameofthron.es$/}
})
< 22841
test>
```

El resultat és que hi ha **22.841** comentaris d'aquest/a usuari/ària amb aquest criteri.

- Exercici 2

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

Fet directament per consola el resultat ens dona que hi ha **3** cinemes. Hem filtrat els documents per incloure només els teatres ubicats a l'estat "DC" i grupat pel codi postal (zipcode). En últim lloc, comptem quants teatres hi ha a cada codi postal:

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

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.

Troblem **4** pel·lícules que compleixen aquests criteris. També mostrem els títols d'aquestes pel·lícules:

```
>_MONGOSH

> db.movies.find(
  {$and:[
    {directors:"John Landis"},
    {"imdb.rating":{$gte:7.5, $lte:8}}
  ]
}
).count()
< 4
test>
```

```
>_MONGOSH

> db.movies.find(
  {
    directors: "John Landis",
    "imdb.rating": { $gte: 7.5, $lte: 8 }
  },
  { title: 1, _id: 0 }
).pretty()
< {
  title: 'Animal House'
}
{
  title: 'The Blues Brothers'
}
{
  title: 'An American Werewolf in London'
}
{
  title: 'Trading Places'
}
test>
```

Amb *Compass* tenim exactament el mateix resultat i també podem veure tota la informació de les quatre pel·lícules:

Documents 23.5K Aggregations Schema Indexes 1 Validation

▼ \$match \$match

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE

```
1 [
2   {
3     $match:
4       /**
5        * query: The query in MQL.
6        */
7     {
8       directors: "John Landis"
9     }
10  },
11  {
12    $match:
13      /**
14       * query: The query in MQL.
15       */
16    {
17      "imdb.rating": {
18        $gte: 7.5,
19        $lte: 8
20      }
21    }
22  }
23 ]
```

PIPELINE OUTPUT

Sample of 4 documents

```
_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/MV5BM2M2ZDA4MTYtOGZjMi00OTg5LWl1ZT..."
  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
```

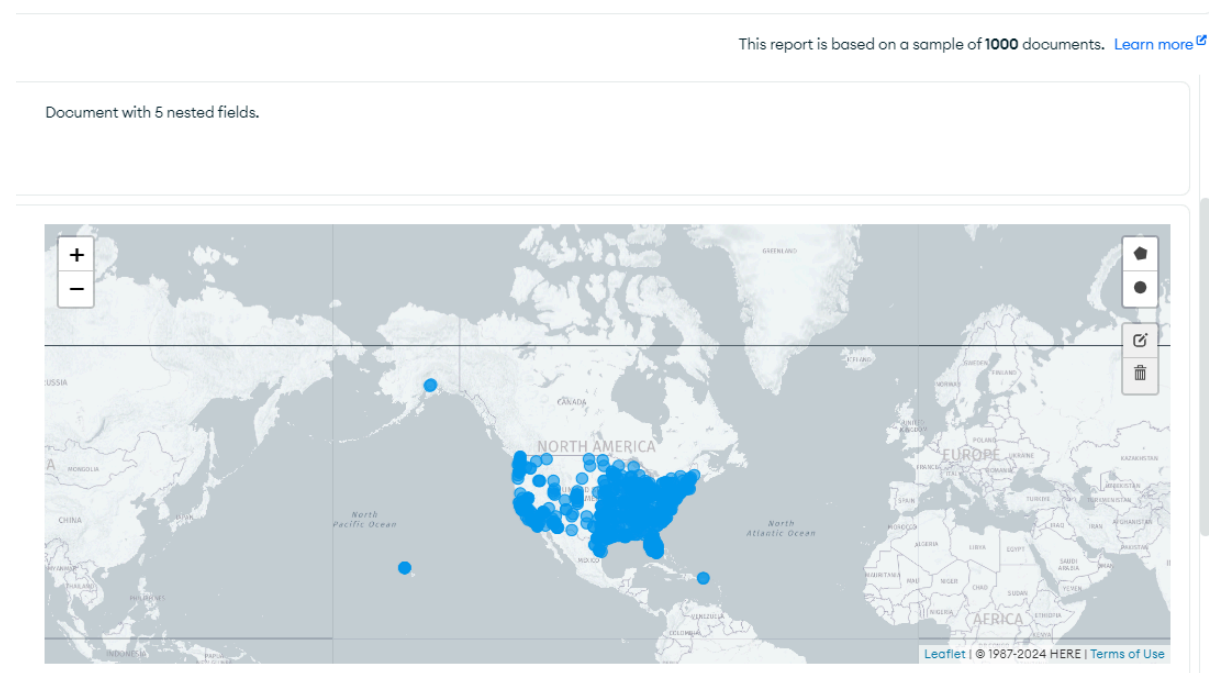
```
_id: ObjectId('573a1397f29313caabce76f7')
plot: "Jake Blues, just out from prison, puts together his old band to save t..."
genres: Array (3)
runtime: 133
rated: "R"
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYTdLMDEExOGU0N2I3MS00MjY5LWl1ZT..."
title: "The Blues Brothers"
fullplot: "After the release of Jake Blues from prison, he and brother Elwood go ..."
languages: Array (1)
released: 1980-06-20T00:00:00.000+00:00
directors: Array (1)
writers: Array (2)
awards: Object
lastupdated: "2015-09-01 00:09:39.130000000"
year: 1980
imdb: Object
```


- Exercici 2

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

Per mostrar el mapa, hem d'anar a la col·lecció **theaters** i si mirem l'esquema veiem que dins de **location** trobem **geo** que mostra un mapa, perquè és de tipus coordenades. Alhora dins hi té **coordinates** que mostra el mateix mapa.

Faig una captura d'aquesta mapa on es mostra la ubicació d'una part dels teatres, ja que si llegim l'avís de la part superior dreta, diu que només veiem una mostra basada en 1000 documents.



Des d'Aggregations si fem un recompte, surt que tenim 1564 teatres, per la qual cosa no es pinten 564 teatres. Tan sols estaríem mostrant un 64% del total.

+

Stage 1 ☒

```
1 ▾ /**
2  * Provide the field name for the count.
3  */
4  'geo'
```

Output after [\\$count](#) stage (Sample of 1 document)

geo : 1564

Opino que amb aquesta mostra del mapa d'ubicacions, és prou rellevant per tenir una idea de la distribució dels teatres de la base de dades.