

# SPRINT\_5

José Messias Garcia da Silva Ferreira

# Exercici 1 - Nivel 1

<<<<



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

```
}  
]  
my_store> db.comments.find().limit(2)  
[  
  {  
    _id: ObjectId('5a9427648b0beebe69579cc'),  
    name: 'Andrea Le',  
    email: 'andrea_le@fakegmail.com',  
    movie_id: ObjectId('573a1390f29313caabcd418c'),  
    text: 'Rem officiiis eaue repellendus amet eos doloribus. Porro dolor voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.',  
    date: ISODate('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 nihil vero animi.',  
    date: ISODate('1987-02-10T00:29:36.000Z')  
  }  
]  
my_store>
```

>>>>

+  
+

# Exercici 1 - Nivel 1

&lt;&lt;&lt;&lt;&lt;



🕒 ▼ {}

✦ Explain Reset Find </> Options ▼

Project { field: 0 }

Sort { field: -1 } or [['field', -1]] Max Time MS 60000

Collation { locale: 'simple' } Skip 0 Limit 2

Index Hint { field: -1 }

⊕ ▼ 📄 ▼ ✎ 🗑️

25 ▼ 1 - 2 of 2 ↺ ⏪ ⏩ ▼ ☰ {} 🗃️

```
_id: ObjectId('5a9427648b0beeb69579cc')
name: "Andrea Le"
email: "andrea_le@fakegmail.com"
movie_id: ObjectId('573a1390f29313caabcd418c')
text: "Rem officiis eaque repellendus amet eos doloribus. Porro dolor volupta..."
date: 2012-03-26T23:20:16.000+00:00
```

```
_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. Co..."
date: 1987-02-10T00:29:36.000+00:00
```

&gt;&gt;&gt;&gt;&gt;



# Exercici 1 - Nivel 1

<<<<<



Quants usuaris tenim registrats?

```
}  
]  
my_store> db.users.countDocuments()  
[  
  185  
]  
my_store> █
```

sprint\_5 > my\_store > users Open MongoDB shell

Documents **185** Aggregations Schema Indexes **1** Validation

🕒 { } 🌟 Explain Reset Find </> Options ▶

+ 📄 ✎ 🗑️ 25 1 - 25 of 185 ↺ ↻ ⌵ ☰ { } 📊

```
{  
  "_id": ObjectId('59b99db4cfa9a34dcd7885b6'),  
  "name": "Ned Stark",  
  "email": "sean_bean@gameofthron.es"  
}
```

>>>>>



# Exercici 1 - Nivel 1

<<<<<

Quants cinemes hi ha en l'estat de Califòrnia?

```
}  
}  
my_store> db.theaters.countDocuments({ "location.address.state": "CA" })  
[  
  169  
]  
my_store>
```

The screenshot shows the MongoDB Compass interface. At the top, there are tabs for Documents (1.6K), Aggregations, Schema, Indexes (1), and Validation. The Documents tab is active. Below the tabs, there is a search bar with a clock icon and a dropdown arrow. The search criteria is `{ "location.address.state": "CA" }`. To the right of the search bar are buttons for Explain, Reset, Find (highlighted in green), and a code icon. Further right is an Options link with a right-pointing triangle. Below the search bar, there is a toolbar with icons for adding documents, a dropdown arrow, a link icon, a pencil icon, a trash icon, and a lightbulb icon. To the right of these icons is a dropdown menu showing 25, followed by the text 1 - 25 of 169. Below the toolbar, there is a list of documents. The first document is expanded, showing its fields: `_id` (ObjectId('59a47286cfa9a3a73e51e72e')), `theaterId` (1008), and `location` (Object). To the right of the document list, there are icons for editing, liking, sharing, and deleting.

>>>>>

++  
++

◀◀◀◀◀



>>>>>

```
my_store> db.users.find().sort({ _id: 1 }).limit(1)
[
  [
    {
      _id: ObjectId('59b99db4cfa9a34dcd7885b6'),
      name: 'Ned Stark',
      email: 'sean_bean@gameofthron.es',
      password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu'
    }
  ]
]
my_store>
```

⌕

▼

{ }

✦

Explain

Reset

Find

</>

Options ▼

Project

{ field: 0 }

Sort

{ "\_id": 1 }

Max Time MS

60000

Collation

{ locale: 'simple' }

Skip

0

Limit

1

Index Hint

{ field: -1 }

+ ▼

🔗 ▼

✎

🗑️

25 ▼

1 - 1 of 1

↺

➡

▼

☰

{ }

📄

```
_id: ObjectId('59b99db4cfa9a34dcd7885b6')  
name: "Ned Stark"  
email : "sean_bean@gameofthron.es"  
password : "$2b$12$UREFwsRUoyF0CRqGNK0LzO0HM/jLhgUCNNIJ9RJAqMUQ74cr1J1Vu"
```

# Exercici 1 - Nivel 1

<<<<<



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

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

Documents 23.5K Aggregations Schema Indexes 1 Validation

🕒 { "genres": "Comedy" }

⚙ Explain Reset Find </> Options ▶

+

🔗

✎

🗑

💡

25 1 - 25 of 7024 ↺ < > ▾ ☰ {} 📄

▶

```
_id: ObjectId('573a1390f29313caabcd4803')  
plot: "Cartoon figures announce, via comic strip balloons, that they will mov..."
```

✎

🔗

📄

🗑

>>>>>

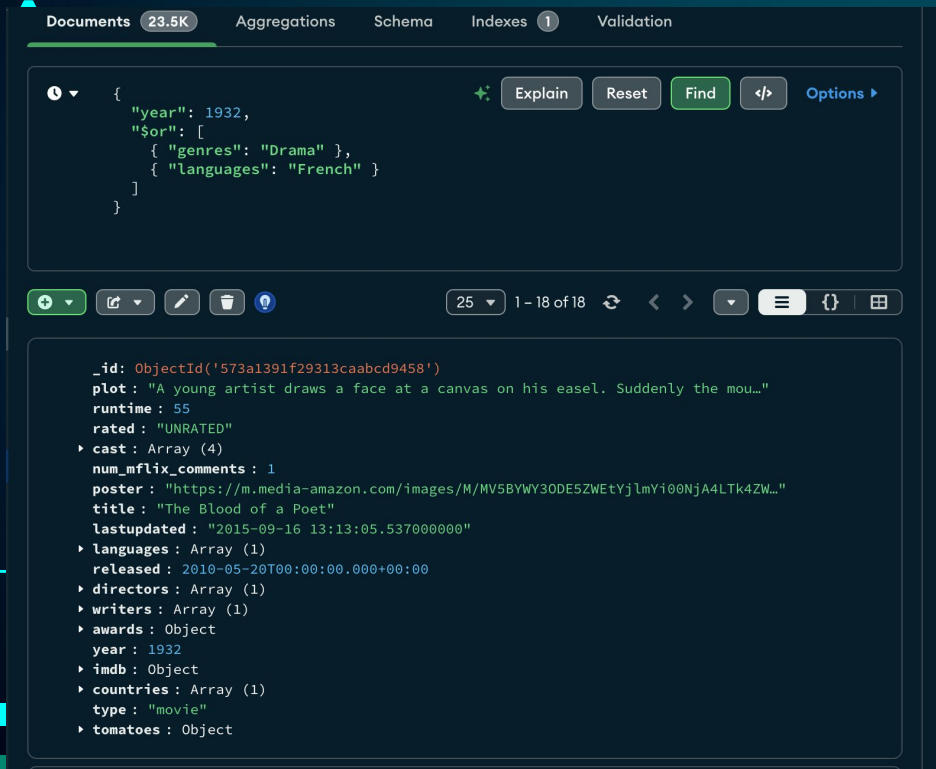




# Exercici 2 - Nivel 1

<<<<

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

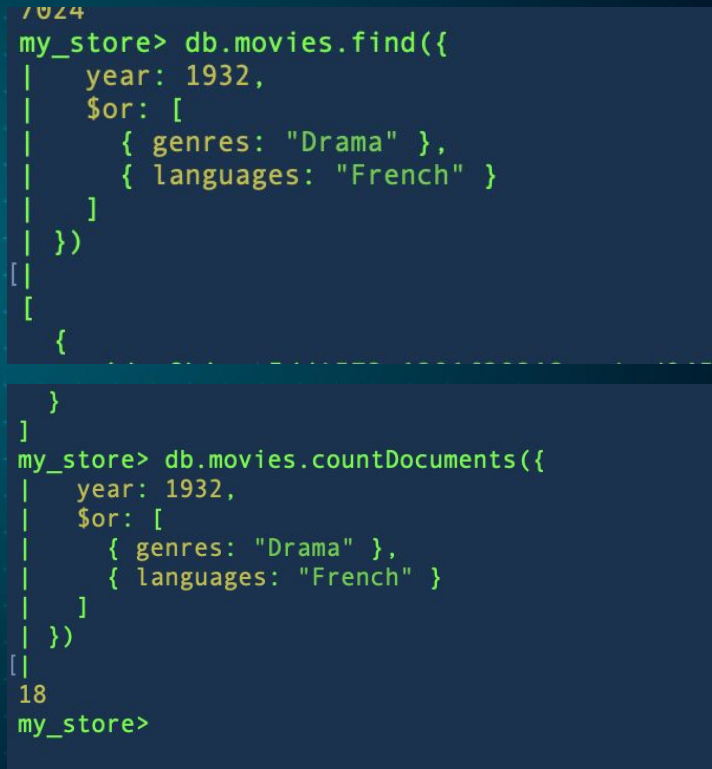


The screenshot shows the MongoDB Compass interface. At the top, there are tabs for Documents (23.5K), Aggregations, Schema, Indexes (1), and Validation. The main area displays a query in JSON format:

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

Below the query, there are buttons for Explain, Reset, Find, and a code icon. The results section shows a single document for the movie "The Blood of a Poet":

```
{
  "_id": ObjectId('573a1391f29313caabdc9458'),
  "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/MV5BYWY30DE5ZWetYjlmYi00NjA4LTk4ZW...",
  "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),
  "awards": Object,
  "year": 1932,
  "imdb": Object,
  "countries": Array (1),
  "type": "movie",
  "tomatoes": Object
}
```



The screenshot shows a MongoDB shell session. The first command is a find query:

```
my_store> db.movies.find({
  |   year: 1932,
  |   $or: [
  |     { genres: "Drama" },
  |     { languages: "French" }
  |   ]
  | })
[
  {
    ...
  }
]
```

The second command is a countDocuments query:

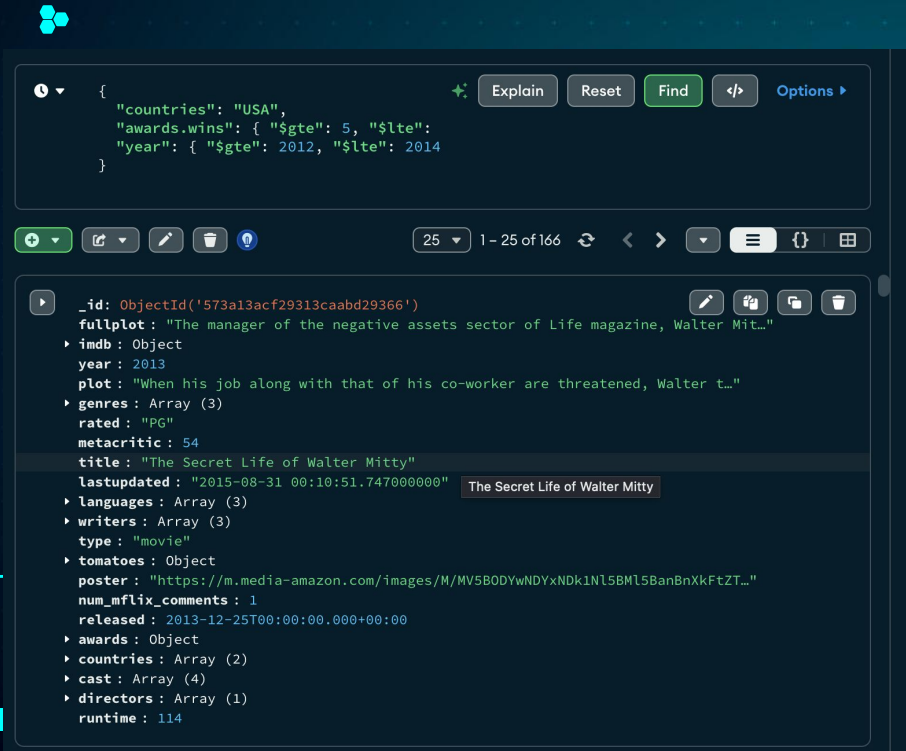
```
my_store> db.movies.countDocuments({
  |   year: 1932,
  |   $or: [
  |     { genres: "Drama" },
  |     { languages: "French" }
  |   ]
  | })
18
my_store>
```



# Exercici 3 - Nivel 1

<<<<

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



The image shows the MongoDB Compass interface. At the top, a query is entered in the JSON editor:

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

Below the query editor, there are buttons for 'Explain', 'Reset', 'Find', and 'Options'. The 'Find' button is highlighted in green. Below these buttons, there are icons for adding, editing, deleting, and refreshing, along with a dropdown menu showing '25' items and a range '1 - 25 of 166'. The main area displays the result of the query, which is a single document for the movie 'The Secret Life of Walter Mitty'.

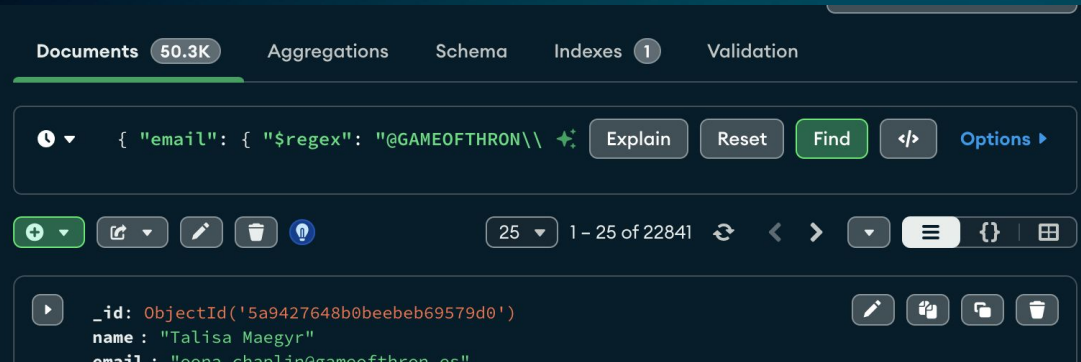
```
{
  "_id": ObjectId('573a13acf29313caabd29366'),
  "fullplot": "The manager of the negative assets sector of Life magazine, Walter Mit...",
  "imdb": Object,
  "year": 2013,
  "plot": "When his job along with that of his co-worker are threatened, Walter t...",
  "genres": Array (3),
  "rated": "PG",
  "metacritic": 54,
  "title": "The Secret Life of Walter Mitty",
  "lastupdated": "2015-08-31 00:10:51.747000000",
  "languages": Array (3),
  "writers": Array (3),
  "type": "movie",
  "tomatoes": Object,
  "poster": "https://m.media-amazon.com/images/M/MV5BODYwNDYxNDk1N15BM15BanBnXkFtZT...",
  "num_mflix_comments": 1,
  "released": 2013-12-25T00:00:00.000+00:00,
  "awards": Object,
  "countries": Array (2),
  "cast": Array (4),
  "directors": Array (1),
  "runtime": 114
}
```

```
18
my_store> db.movies.find({
|   countries: "USA",
|   "awards.wins": { $gte: 5, $lte: 9 },
|   year: { $gte: 2012, $lte: 2014 }
| })
[
|
```

# Exercici 1 - Nivel 2

<<<<

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



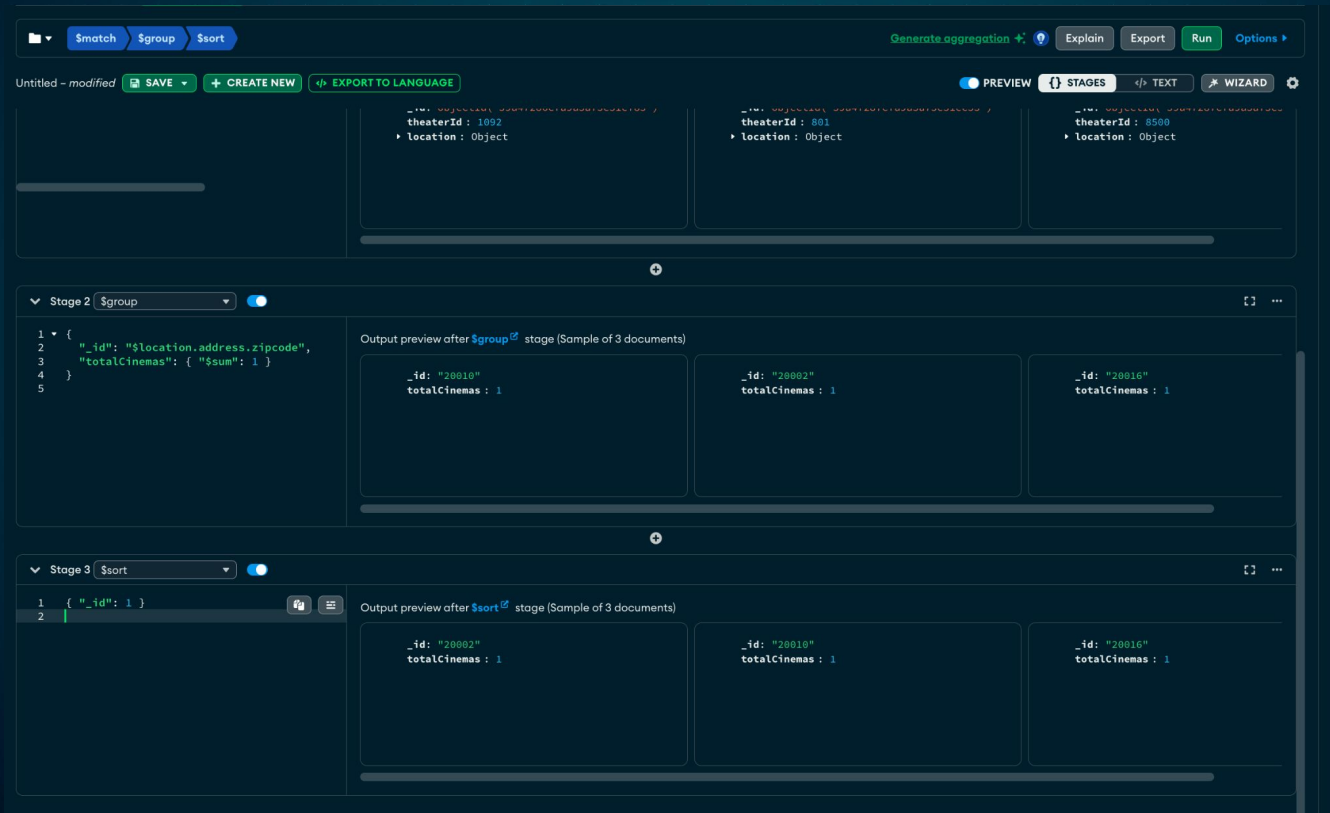
```
22841
my_store> db.comments.countDocuments({
|   "email": { "$regex": "gameofthron", "$options": "i" }
| })
[
22841
my_store>
```

>>>>

# Exercici 2 - Nivel 2

<<<<

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



The screenshot shows a data pipeline configuration in a web interface. The pipeline consists of three stages: **\$match**, **\$group**, and **\$sort**. The **\$group** stage is expanded, showing a sample of 3 documents and their aggregation results.

**Stage 1: \$match**

- Filter: `location.address.zipcode == "20010"`
- Output preview after **\$match** stage (Sample of 3 documents):

Document
<code>{ "theaterId": 1092, "location": { "address": { "zipcode": "20010" } } }</code>
<code>{ "theaterId": 881, "location": { "address": { "zipcode": "20010" } } }</code>
<code>{ "theaterId": 8500, "location": { "address": { "zipcode": "20010" } } }</code>

**Stage 2: \$group**

- Group by: `location.address.zipcode`
- Aggregation: `$sum: 1`
- Output preview after **\$group** stage (Sample of 3 documents):

Document
<code>{ "_id": "20010", "totalCinemas": 1 }</code>
<code>{ "_id": "20002", "totalCinemas": 1 }</code>
<code>{ "_id": "20016", "totalCinemas": 1 }</code>

**Stage 3: \$sort**

- Sort by: `totalCinemas`
- Output preview after **\$sort** stage (Sample of 3 documents):

Document
<code>{ "_id": "20002", "totalCinemas": 1 }</code>
<code>{ "_id": "20010", "totalCinemas": 1 }</code>
<code>{ "_id": "20016", "totalCinemas": 1 }</code>

# Exercici 2 - Nivel 2

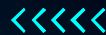
<<<<<

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

```
{ _id: '20016', totalCinemas: 1 }  
]  
my_store> db.theaters.aggregate([  
|   {  
|     $match: { "location.address.state": "DC" }  
|   },  
|   {  
|     $group: {  
|       _id: "$location.address.zipcode",  
|       totalCinemas: { $sum: 1 }  
|     }  
|   },  
|   { $sort: { _id: 1 } } // opcional: ordena per codi postal  
| ])  
[  
| [  
|   { _id: '20002', totalCinemas: 1 },  
|   { _id: '20010', totalCinemas: 1 },  
|   { _id: '20016', totalCinemas: 1 }  
| ]  
]  
my_store>
```

>>>>>

# Exercici 1 - Nivel 3



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



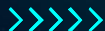
The screenshot shows a MongoDB query interface. The query is:

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

The interface includes fields for Project, Sort, Collation, Index Hint, Max Time MS, Skip, and Limit. The results are displayed in a table with columns for \_id, plot, genres, runtime, rated, cast, num\_mflix\_comments, poster, title, fullplot, languages, released, directors, writers, awards, lastupdated, year, imdb, countries, type, and tomatoes.

The first result is for the movie "The Blues Brothers":

```
{
  "_id": ObjectId('573a1397f29313caabce76ff7'),
  "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/MV5BYTd1MDExOGU2N2I3MS00MjY5LWE1NT...",
  "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,
  "countries": Array (1),
  "type": "movie",
  "tomatoes": Object
}
```



# Exercici 1 - Nivel 3

<<<<

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



```
}
my_store> db.movies.find(
| {
|   "directors": "John Landis",
|   "imdb.rating": { $gte: 7.5, $lte: 8 }
| },
| { "title": 1, "directors": 1, "imdb.rating": 1, "_id": 0 }
| )
[|
| [
|   {
|     imdb: { rating: 7.6 },
|     title: 'Animal House',
|     directors: [ 'John Landis' ]
|   },
|   {
|     title: 'The Blues Brothers',
|     directors: [ 'John Landis' ],
|     imdb: { rating: 7.9 }
|   },
|   {
|     imdb: { rating: 7.6 },
|     title: 'An American Werewolf in London',
|     directors: [ 'John Landis' ]
|   },
|   {
|     title: 'Trading Places',
|     directors: [ 'John Landis' ],
|     imdb: { rating: 7.5 }
|   }
| ]
| ]
my_store>
```

>>>>

+  
+



# Exercici 2 - Nivel 3

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

sprint\_5 > my\_store > theaters [Open MongoDB shell](#)

Documents 1.6K Aggregations Schema Indexes 3 Validation

**Query Builder**

Project: { field: 0 }

Sort: { field: -1 } or [['field', -1]] Max Time MS: 40000

Collation: { locale: 'simple' } Skip: 0 Limit: 0

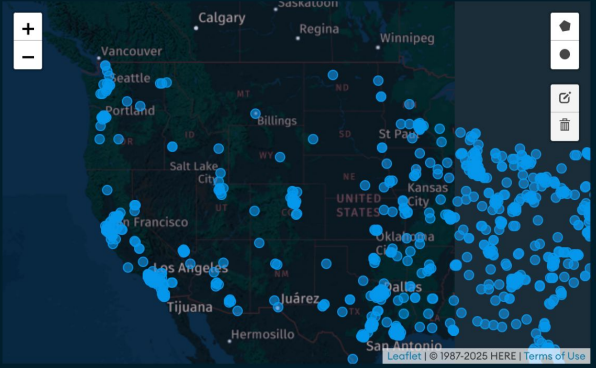
Index Hint: { field: -1 }

[EXPORT SCHEMA](#)

This report is based on a sample of 1000 documents. [Learn more](#)

**geo**

coordinates



**theaterId**

int32

1540	2857	36	2866	350	831	241	2865	1149	447	102	2792	1793
1513	1193	2982	2977	1794	474	1064						



