

Sprint 5

Bases de dates amb MongoDB

Borja Munitiz

Nivell 1

Exercici 1

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

Para ver los dos comentarios primeros comentarios de la base de datos en las opciones de visualización de la pestaña de QUERIES podemos limitar la vista a 2 entradas, suponiendo que se quieren ver por orden de inserción, en caso de querer hacerlo por orden cronológico habría que ordenarlo primero por el campo DATE

MongoDB > Sprint5 > comments

Documents 50.3K Aggregations Schema Indexes 1 Validation

Find

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

Project { field: 0 }

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

Max Time MS 60000

Collation { locale: 'simple' }

Skip 0

Limit 2

Index Hint { field: -1 }

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

25 1 - 2 of 2

Document 1:

```
_id: ObjectId('5a9427648b0beebeb69579cc')
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
```

Document 2:

```
_id: ObjectId('5a9427648b0beebeb69579cf')
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
```

Nivell 1

Exercici 1

Quants usuaris tenim registrats?

Para saber el número de usuarios registrados accedemos a la pestaña de AGGREGATIONS y creamos un STAGE en la tabla USERS donde podemos generar un COUNT a partir del campo NAME para no repetir usuarios y nos da el número de usuarios.

The screenshot shows the MongoDB Aggregations interface for the 'users' collection. The 'Aggregations' tab is selected. A single stage, '\$count', is defined in the pipeline. The preview section shows three sample documents from the collection:

```
_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name : "Ned Stark"
email : "sean_bean@gomeofthron.es"
password : "$2b$12$UREFwsRUoyF0CRqGNK0Lz00H..."

_id: ObjectId('59b99db4cfa9a34dcd7885b7')
name : "Robert Baratheon"
email : "mark_addy@gomeofthron.es"
password : "$2b$12$yGqxLG9LZpXA2xVDhuPnSOZd...

_id: ObjectId('59b99db5cfa9a34dcd7885b8')
name : "Jaime Lannister"
email : "nikolaj_coster-
walda@gomeofthron.es
password : "$2b$12$6vz7wiw.O.E"
```

The Stage 1 panel shows the aggregation pipeline stage:

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

The output after the \$count stage shows a sample document with a single field:

```
name : 185
```

Nivell 1

Exercici 1

Quants cinemes hi ha en l'estat de Califòrnia?

Para saber la cantidad de cines en el estado de California creamos un AGGREGATION PIPELINE en la tabla THEATERS donde creamos un STAGE para filtrar los cines del estado de California y a continuación un STAGE con COUNT para saber la cantidad.

The screenshot shows the MongoDB Aggregations interface with the following details:

- Documents:** 1.6K
- Aggregations:** (selected)
- Schema**
- Indexes:** 1
- Validation**

The pipeline consists of two stages:

- Stage 1 (\$match):** Filters documents where "location.address.state" is "CA".
- Stage 2 (\$count):** Provides the field name for the count, specifically "theaterId".

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

```
_id: ObjectId('59a47286cfa9a3a73e51e72e')
theaterId : 1008
location : Object
```

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

```
theaterId : 169
```

Nivell 1

Exercici 1

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

Para saber qué usuario/usuaria se registró primero asumimos que la fecha correspondiente al primer comentario es la fecha en la que se inscribió el usuario/usuaria, para ello creamos un AGGREGATION PIPELINE con un primer STAGE en el que ordenamos por DATE y continuación otro STAGE en el que limitamos al primer resultado, podemos observar que la fecha corresponde a 1970 lo cual no debería darse, entendemos que al ser una base de datos de prueba los datos pueden no corresponderse con la realidad.

The screenshot shows the MongoDB Aggregations interface. The top navigation bar includes 'Documents' (50.3K), 'Aggregations' (selected), 'Schema', 'Indexes' (1), and 'Validation'. Below the bar are buttons for '\$sort' and '\$limit'. On the right, there are buttons for 'Generate aggregation', 'Preview', 'Stages', 'Text', 'WIZARD', and 'Run'. The main area is titled 'Untitled - modified'. It shows a pipeline with two stages:

- Stage 1 (\$sort)**:
A code editor containing the following pipeline stage:

```
1  /**  
2   * Provide any number of field/order pair  
3   */  
4  {  
5    date: 1  
6  }
```
- Output after \$sort stage (Sample of 10 documents)**:
A list of 10 documents. The first document is shown in full:

```
_id: ObjectId('5a9427648b0beebeb695e67e')  
name : "Mercedes Tyler"  
email : "mercedes_tyler@fakegmail.com"  
movie_id : ObjectId('573a1399f29313caabcec3c...  
text : "Optio totam dolores magni. Enim  
ratione fuga tempora voluptatum est  
cu..."  
date : 1970-01-01T01:07:09.000+00:00
```
- Stage 2 (\$limit)**:
A code editor containing the following pipeline stage:

```
1  /**  
2   * Provide the number of documents to lim  
3   */  
4  1
```
- Output after \$limit stage (Sample of 1 document)**:
A list showing the single document from the previous stage:

```
_id: ObjectId('5a9427648b0beebeb695e67e')  
name : "Mercedes Tyler"  
email : "mercedes_tyler@fakegmail.com"  
movie_id : ObjectId('573a1399f29313caabcec3c...  
text : "Optio totam dolores magni. Enim  
ratione fuga tempora voluptatum est  
cu..."
```

Nivell 1

Exercici 1

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

Para contabilizar la cantidad de películas de comedia hay en la base de datos generamos una AGGREGATION PIPELINE en la tabla MOVIES, en el primer STAGE filtramos por género para quedarnos con aquellas películas contengan el campo COMEDY y continuación hacemos un conteo de los resultados.

The screenshot shows the MongoDB Aggregation Pipeline interface. At the top, there are tabs for 'Documents' (23.5K), 'Aggregations' (selected), 'Schema', 'Indexes' (1), and 'Validation'. Below the tabs, there are buttons for '\$match' and '\$count'. On the right side, there are buttons for 'Generate aggregation', 'Preview', 'Stages', 'Text', 'Wizard', 'Run', and 'Options'. The main area is divided into two sections: 'Stage 1 \$match' and 'Stage 2 \$count'. Stage 1 contains the following MQL code:

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

The output after the \$match stage shows sample documents:

```
_id: ObjectId('573a1390f29313caabcd4803')
plot: "Cartoon figures announce, via comic strip balloons, that they will mov..."
genres: Array (3)
runtime: 7
cast: Array (1)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYzg2NjNhNT..."
```

Stage 2 contains the following MQL code:

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

The output after the \$count stage shows the result:

```
string: 7024
```

Nivell 1

Exercici 2

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

Para mostrar los documentos de películas producidas en 1932 y que sean del género drama o estén en francés creamos una QUERY en la tabla MOVIES donde utilizamos el operador \$AND para asegurarnos que se cumplen las todas las condiciones, para verificar que se cumple una de las dos condiciones de lenguaje o género usamos un operador \$OR dentro del operador \$AND, de esta forma nos aseguramos que siempre se cumplan las condiciones indispensables (año = 1932 y (lenguaje = francés o género = drama)

The screenshot shows a MongoDB interface with the following details:

- Documents:** 23.5K
- Aggregations:** 1
- Schema:** 1
- Indexes:** 1
- Validation:** 1

The query bar contains the following expression:

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

Below the query bar are buttons for **Generate query**, **Explain**, **Reset**, **Find**, and **Options**.

The results section displays two movie documents:

```
_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)
awards: Object
year: 1932
imdb: Object
countries: Array (1)
type: "movie"
tomatoes: Object
```



```
_id: ObjectId('573a1392f29313caabcd99a3')
plot: "Junta is hated by the people in the village where she lives, especiall..."
genres: Array (3)
runtime: 85
cast: Array (4)
poster: "https://m.media-amazon.com/images/M/MV5BNTQ1NTMzMTQtODIyYS00MTAxLWE1NT..." 
title: "The Blue Light"
fullplot: "Junta is hated by the people in the village where she lives, especiall..."
languages: Array (2)
released: 1934-05-08T00:00:00.000+00:00
```

Nivell 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

Para mostrar las películas estadounidenses que tengan entre 5 y 9 premios y que hayan sido producidas entre 2012 y 2014 utilizaremos una QUERY con el operador \$AND donde filtramos por COUNTRIES y los operadores \$GTE (greater than or equal) y \$LTE (less than or equal) para filtrar los rangos de premios y años.

The screenshot shows a MongoDB interface with the following details:

- Documents:** 23.5K
- Aggregations:** 1
- Schema:** 1
- Indexes:** 1
- Validation:** 1

Query:

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

Buttons: ADD DATA, EXPORT DATA, UPDATE, DELETE, INSIGHT, Generate query, Explain, Reset, Find (highlighted), Options.

Results: 25 - 1 - 25 of 166

Document 1:

```
_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/MV5BODYwNDYxNDk1Nl5BMl5BanBnXkFtZT... "
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
```

Document 2:

```
_id: ObjectId('573a13b5f29313caabd45772')
plot: "After their cave is destroyed, a caveman family must trek through an u..."
genres: Array (3)
```

Nivell 2

Exercici 1

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

Para contar los comentarios de un usuario/usuaria que utiliza "GAMEOFTHRON.ES" como dominio de correo electrónico, como el enunciado no clarifica si se quiere saber la cantidad de comentarios por cada usuario/usuaria que utiliza ese correo electrónico o el total de comentarios escritos por los usuarios/usuarias que utilizan dicho correo, he optado por contar el total de comentarios escritos por usuarios/usuarias de ese correo, para ello he creado un AGGREGATION PIPELINE donde el primer STAGE filtra la tabla USERS con el operador \$MATCH y \$REGEX para comparar solo la parte final de las direcciones de correo electrónico, después se crea un segundo STAGE con \$COUNT para tener el número total.

The screenshot shows the MongoDB Aggregations interface with the following details:

- Documents:** 50.3K
- Aggregations:** Selected tab
- Schema:** Indexes: 1
- Validation:**

Stages:

- Stage 1: \$match**

```
1 /**  
2  * query: The query in MQL.  
3 */  
4 {  
5   email: {$regex: /GAMEOFTHRON.ES$/i}  
6 }
```
- Output after \$match stage (Sample of 10 documents):**

```
_id: ObjectId('5a9427648b0beeb69579d0')  
name : "Talisa Maegyr"  
email : "oona_chaplin@gameofthron.es"  
movie_id : ObjectId('573a1390f29313caabcd41b...  
text : "Rem itaque ad sit rem voluptatibus.  
Ad fugiat maxime illum optio iure ..."  
date : 1998-08-22T11:45:03.000+00:00
```
- Stage 2: \$count**

```
1 /**  
2  * Provide the field name for the count.  
3 */  
4 'name'|
```
- Output after \$count stage (Sample of 1 document):**

```
name : 22841
```

Buttons at the top right: Generate aggregation, Explain, Export, Run, Options.

Nivell 2

Exercici 2

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

Para contar cuantos cines hay por código postal en Washington DC, generamos un AGGREGATION PIPELINE donde filtramos con el operador \$MATCH para saber los cines de Washington DC y a continuación creamos otro STAGE donde agrupamos por código postal y contamos cuantos cines hay en cada uno, esto se consigue con el operador \$GROUP y dentro de este un \$COUNT.

The screenshot shows the MongoDB Aggregations interface for the 'theatres' collection in the 'Sprint5' database. The pipeline consists of two stages:

- Stage 1 (\$match):** Filters documents where the state is "DC". The output sample shows three documents, each representing a theater located in Washington D.C. with a specific address and zip code.
- Stage 2 (\$group):** Groups documents by zipcode and counts the number of theaters per group. The output sample shows two groups: one for zipcode "20016" (count 1) and one for zipcode "20002" (count 1).

```
1 /**  
2 * query: The query in MQL.  
3 */  
4 {  
5   "location.address.state": "DC"  
6 }  
  
1 _id: ObjectId('59a47287cfa9a3a73e51ec33')  
theaterId: 801  
location: Object  
address: Object  
street1: "4500 Wisconsin Ave Nw"  
city: "Washington"  
state: "DC"  
zipcode: "20016"  
geo: Object  
  
1 _id: ObjectId('59a47287cfa9a3a73e51ec3d8')  
theaterId: 8500  
location: Object  
address: Object  
street1: "50 Massachusetts Avenue NE"  
street2: null  
city: "Washington"  
state: "DC"  
zipcode: "20002"  
  
1 _id: "20016"  
count: 1  
1 _id: "20002"  
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

Para encontrar las películas de John Landis con una puntuación IMDB entre 7,5 y 8 hacemos una QUERY con el operador \$AND y los operadores \$EQ (equal) para filtrar por director y \$GTE y \$LTE para filtrar el rango de puntuación.

The screenshot shows a MongoDB query interface with the following details:

- Documents:** 23.5K
- Aggregations:** 1
- Schema:** 1
- Indexes:** 1
- Validation:**

Query:

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

Actions:

- Generate query
- Explain
- Reset
- Find
- Options

Results: 25 1 - 4 of 4

The results show two movie documents:

```
_id: ObjectId('573a1397f29313caabce6d94')
fullplot: "Faber College has one frat house so disreputable it will take anyone. ..."
imdb: Object
  rating: 7.6
  votes: 84834
  id: 77975
  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/MV5BM2M2ZDA4MTYt0GRjMi000Tg5LWI1ZT..."
    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

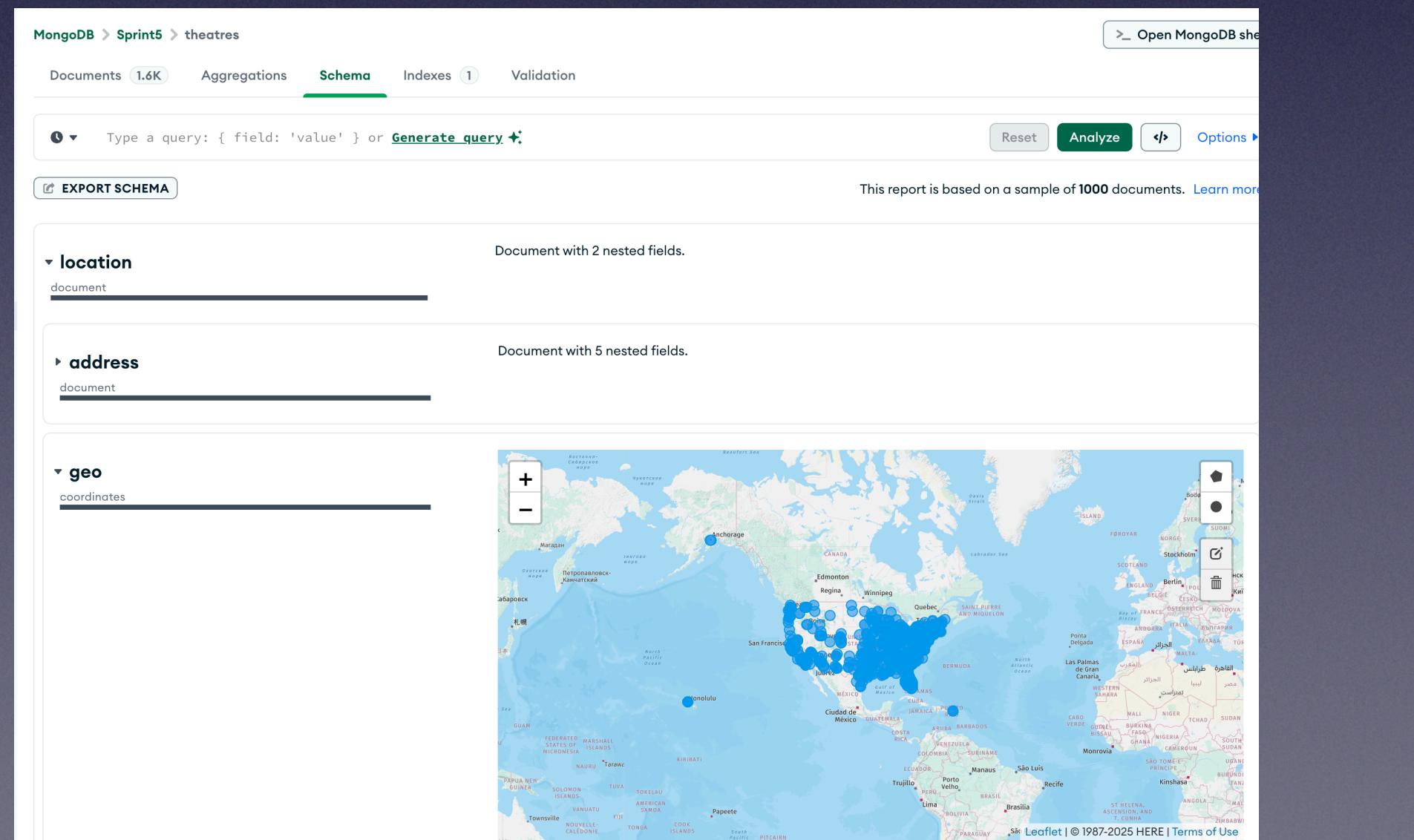
_id: ObjectId('573a1397f29313caabce76f7')
plot: "Jake Blues, just out from prison, puts together his old band to save t..."
genres: Array (3)
runtime: 133
rated: "R"
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYTd1MDExOGUtN2I3MS00MjY5LWE1NT..."
```

Nivell 3

Exercici 2

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

Para mostrar la ubicación de los cines en un mapa accedemos a la pestaña SCHEMA en la tabla THEATERS y con el botón ANALYZE podemos acceder a las coordenadas almacenadas en GEO y en COORDINATES, aquí podemos ver un mapa con las localizaciones de todos los cines.



MongoDB > Sprint5 > theatres

Documents 1.6K Aggregations Schema Indexes 1 Validation

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

EXPORT SCHEMA

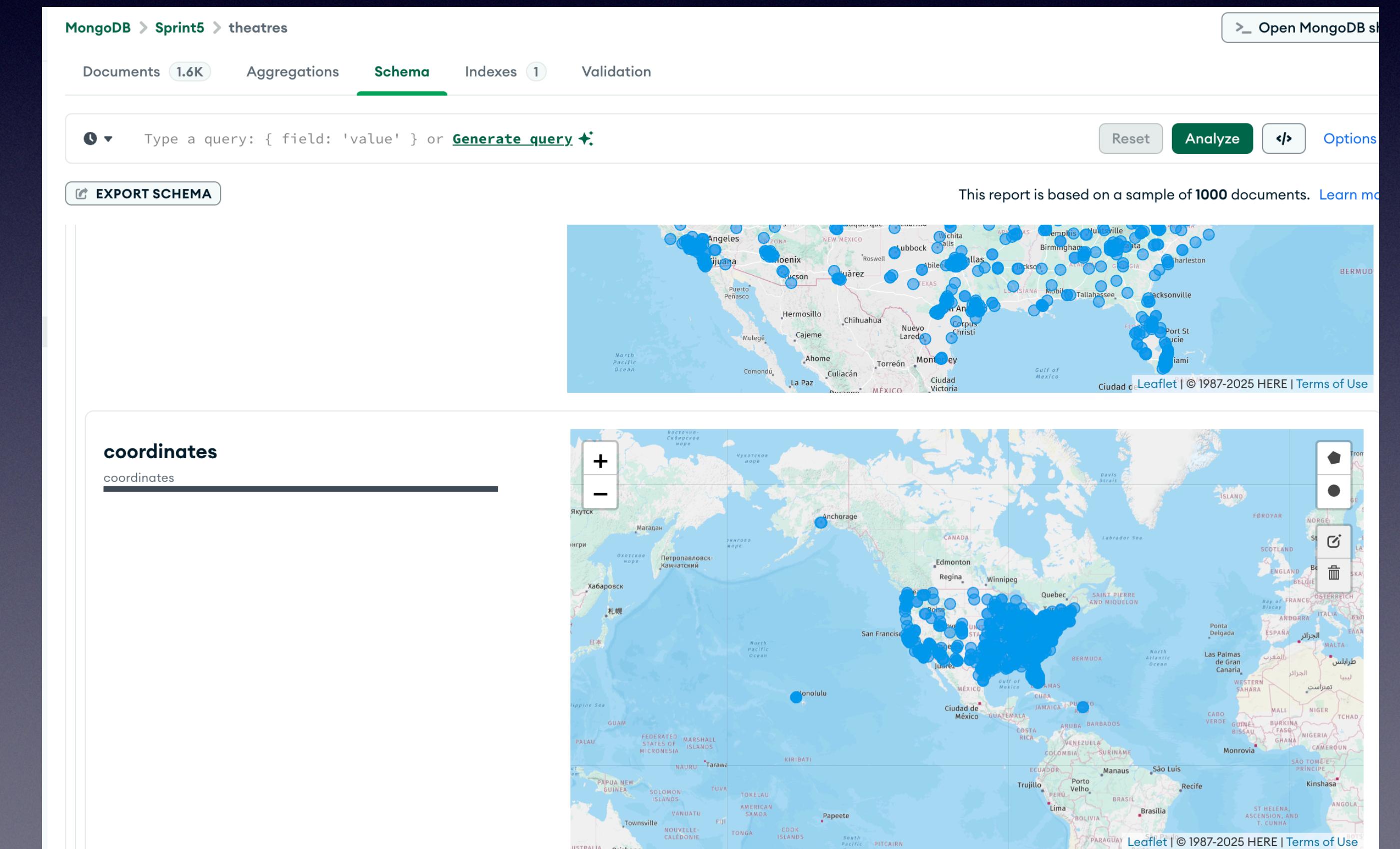
This report is based on a sample of 1000 documents. [Learn more](#)

location document

address document

geo coordinates

coordinates



MongoDB > Sprint5 > theatres

Documents 1.6K Aggregations Schema Indexes 1 Validation

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

EXPORT SCHEMA

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

coordinates

coordinates

Open MongoDB schema

Analyze Options

Reset

Leaflet | © 1987-2025 HERE | Terms of Use

Leaflet | © 1987-2025 HERE | Terms of Use