

## Descripción

Trabajaremos con una base de datos que contiene colecciones relacionadas con una aplicación de entretenimiento cinematográfico:

users: Almacena información de usuarios/as, incluyendo nombres, emails y contraseñas cifradas.

theatres: Contiene datos de cines, como ID, ubicación (dirección y coordenadas geográficas).

sessions: Guarda sesiones de usuario, incluyendo ID de usuario y tokens JWT para la autenticación.

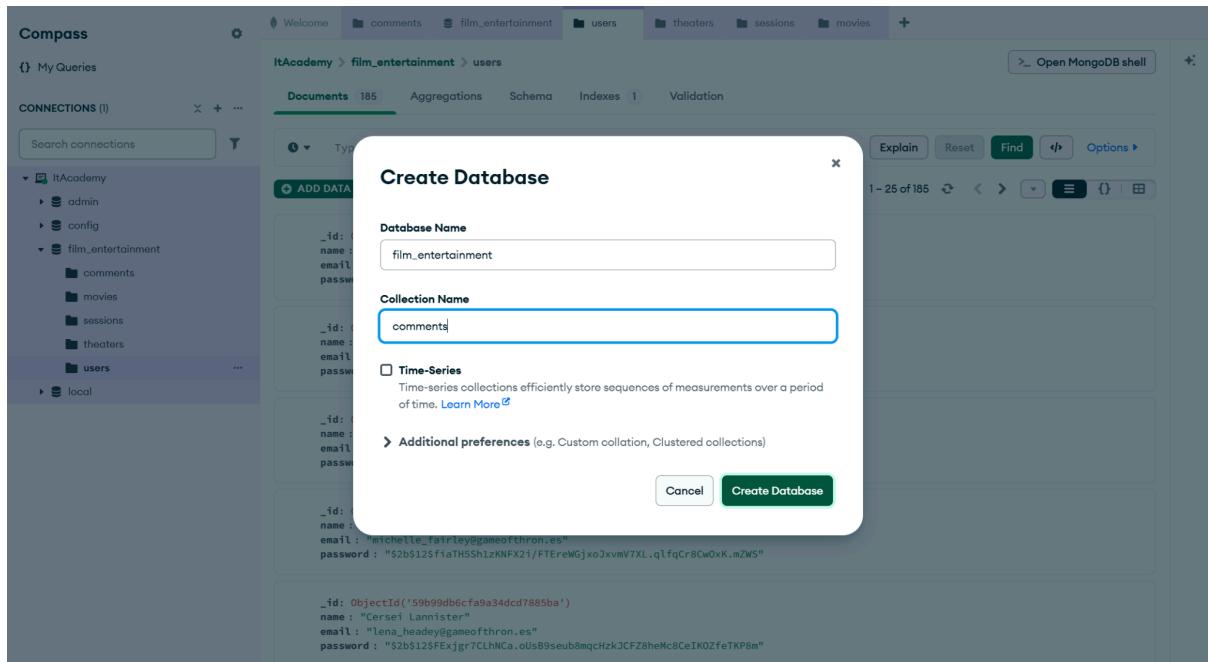
movies: Incluye detalles de películas, como trama, géneros, duración, elenco, comentarios, año de lanzamiento, directores, clasificación y premios.

comments: Almacena comentarios de usuarios/as sobre películas, con información del autor/a del comentario, ID de la película, texto del comentario y la fecha.

Realizarás algunas consultas que te pide el cliente/a, quien está midiendo si serás capaz o no de hacerte cargo de la parte analítica del proyecto vinculado con su base de datos.

## Nivel 1

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



Para crear la base de datos nos dirigimos a la conexión que tenemos activa (en nuestro caso <ItAcademy>) y damos clic en el botón <+>. Una vez dado clic se despliega una ventana <Create Database> aquí tenemos que nombrar a nuestra base de datos <film\_entertainment> y tenemos también que nombrar la primera <collection> que en nuestro caso será el nombre del primer archivo json que queremos importar, <comments>. Y finalmente damos clic a create database.

## B. Agregar collections

The screenshot shows the MongoDB Compass interface. The title bar reads "MongoDB Compass - ItAcademy/film\_entertainment.users". The menu bar includes "Connections", "Edit", "View", "Collection", and "Help". The top navigation bar has tabs for "Welcome", "comments", "film\_entertainment", "theaters", "users" (which is selected), "sessions", "movies", and a "+" button. On the left, the "Compass" sidebar shows "My Queries" and a "CONNECTIONS (1)" section with "ItAcademy" expanded, showing "admin", "config", "film\_entertainment" (which is selected), "comments", "movies", "sessions", "theaters", and "users". Below this is a "local" section. The main area is titled "ItAcademy > film\_entertainment > users". It has tabs for "Documents" (0), "Aggregations", "Schema", "Indexes" (1), and "Validation". A search bar says "Type a query: { field: 'value' } or Generate query". Below it are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". To the right are "Explain", "Reset", "Find" (highlighted in green), and "Options". At the bottom, it says "25 0 - 0 of 0" and has icons for sorting and filtering. A message in the center says "This collection has no data" with a note: "It only takes a few seconds to import data from a JSON or CSV file." A green "Import data" button is at the bottom.

Una vez creada nuestra base de datos y nuestra primera collection, tenemos que importar los datos de esta collection. Y esto lo hacemos simplemente dando clic a <import data> que nos sale en la ventana de la collection creada.

Repetimos este paso para cada una de las collections que tiene nuestra base de datos. Recordando que ahora cada vez que queremos crear una nueva, daremos clic en el <+> desde nuestra base de datos, ya no desde la conexión principal.

### C. Importar datos a collections

The screenshot shows the MongoDB Compass application interface. The title bar reads "MongoDB Compass - ItAcademy/film\_entertainment.users". The top navigation bar includes "Connections", "Edit", "View", "Collection", and "Help". Below the title bar, there's a tabs section with "Welcome", "comments", "film\_entertainment", "theaters", "users" (which is selected), "sessions", and "movies". On the left, a sidebar titled "Compass" shows "My Queries" and "CONNECTIONS (1)". The main area is titled "ItAcademy > film\_entertainment > users" and shows "Documents 185". A search bar says "Type a query: { field: 'value' } or Generate query +". Below it are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". A status bar at the bottom right shows "25 1 - 25 of 185". The main content area lists five document snippets:

```

_id: ObjectId('59b99db4cfa9a34dc7885b6')
name : "Ned Stark"
email : "sean_bean@gameofthron.es"
password : "$2b$12$UREFwsRUioyFOCRqGNK0Lz0@HM/jLhgUCNNI39RJAqMUQ74cr1J1Vu"

_id: ObjectId('59b99db4cfa9a34dc7885b7')
name : "Robert Baratheon"
email : "mark_addy@gameofthron.es"
password : "$2b$12$yGqxLG9LpzXA2xVDhuPnSOZd.VURVkj7wgOLY3pn0s7u2S1Z032y"

_id: ObjectId('59b99db5cfa9a34dc7885b8')
name : "Jaime Lannister"
email : "nikolaj_coster-waldau@gameofthron.es"
password : "$2b$12$6vz7wIwD.E15Rilvq1zUc./9480gb1uPtXcahDxIdagyC3PS8XCUK"

_id: ObjectId('59b99db5cfa9a34dc7885b9')
name : "Catelyn Stark"
email : "michelle_fairley@gameofthron.es"
password : "$2b$12$fa1TH5Sh1zKNFX2f/FTErEWGjxoJxvmV7XL.qlfqCr8CwOxK.mZWS"

_id: ObjectId('59b99db6cfa9a34dc7885ba')
name : "Cercei Lannister"
email : "lena_headley@gameofthron.es"
password : "$2b$12$FExJgr7CLhNCa.oUsB9seub8mqcHzkJCFZ8heMc8CeIK0ZfeTKP8m"

```

A dark overlay at the bottom left says "Import completed. 185 documents imported.".

Y finalmente una vez importados los datos, esta será la visualización que obtenemos. Podemos ver la cantidad de documentos en la parte superior, por encima de ADD DATA, en el caso de users son 185 documentos.

## Ejercicio 1

Muestra los 2 primeros comentarios que aparecen en la base de datos.

The screenshot shows two instances of MongoDB Compass. The top instance is titled "MongoDB Compass - ItAcademy/film\_entertainment.comments" and displays the "comments" collection from the "film\_entertainment" database. A search bar at the top has the query "Project { field: 0 } Sort { field: -1 } or [ { 'field': -1 } ] Collection { locale: 'simple' } Index Hint { field: -1 }". Below the search bar are "Max Time MS" (60000), "Skip" (0), and "Limit" (2) fields. The results table shows two documents:

```
_id: ObjectId('5a9427648bb6beeb669579cc')
name: 'Andrea Le'
email: 'andrea_le@fakemail.com'
movie_id: ObjectId('573a1390f29313caabcd418c')
text: 'Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptate...
date: 2012-03-26T23:20:16.000Z

_id: ObjectId('5a9427648bb6beeb669579cf')
name: 'Greg Powell'
email: 'greg_powell@fakemail.com'
movie_id: ObjectId('573a1390f29313caabcd41b1')
text: 'Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co...
date: 1987-02-10T00:29:36.000Z
```

The bottom instance is titled "MongoDB Compass - ItAcademy/Shell" and shows the MongoDB Shell interface. The command entered is:

```
> use film_entertainment
< switched to db film_entertainment
> db["comments"].find().limit(2)
< [
  {
    _id: ObjectId('5a9427648bb6beeb669579cc'),
    name: 'Andrea Le',
    email: 'andrea_le@fakemail.com',
    movie_id: ObjectId('573a1390f29313caabcd418c'),
    text: 'Rem officiis eaque repellendus amet eos doloribus. Porro dolor voluptate...
    date: 2012-03-26T23:20:16.000Z
  }
  {
    _id: ObjectId('5a9427648bb6beeb669579cf'),
    name: 'Greg Powell',
    email: 'greg_powell@fakemail.com',
    movie_id: ObjectId('573a1390f29313caabcd41b1'),
    text: 'Tenetur dolorum molestiae ea. Eligendi praesentium unde quod porro. Co...
    date: 1987-02-10T00:29:36.000Z
  }
]>
```

Para realizar consultas dentro de COMPASS podemos realizarlas de dos maneras; La primera desde el <tab> de <filter>, y la otra desde Shell de MongoDB.

Al usar el tab, podemos observar que existen unas llaves vacías, las dejamos vacías y damos clic en options, una vez ahí elegimos Limit (2).

Al usar Shell escribimos una línea de comando, `db["comments"].find().limit(2)` en la cual indicamos la collection de la cual queremos hacer la consulta, la acción principal <find()> y la condición <limit(2)>. Ejecutamos y nos devuelve el resultado de la consulta.

¿Cuántos usuarios tenemos registrados?

The screenshot shows the MongoDB Compass interface. In the left sidebar, under 'Connections', the 'ItAcademy' connection is selected. Under 'ItAcademy', the 'users' collection is highlighted. The main area shows the 'Aggregations' tab selected. A pipeline is being built with one stage: '\$count'. The output preview shows a single document with a field 'total\_users' containing the value '185'. At the top right, there are buttons for 'PREVIEW', 'STAGES', 'TEXT', 'WIZARD', 'EXPLAIN', 'EXPORT', 'Run', and 'Options'.

Si queremos resolver esta consulta sin utilizar shell, tenemos que ir a la collection <users> y desde ahí ir a la pestaña Aggregations. Dentro de esta pestaña damos clic a <add stage> y seleccionamos <count>. Debajo de la mini ventana que se genera <Stage-1> ponemos el nombre de este "campo" <total\_users> y lo visualizamos a la derecha 185 users.

También podemos encontrar este dato, de una manera más fácil. Simplemente viendo en la pestaña documents, junto a esta dice un numero, este numero es el total de documentos (en nuestro caso users) que contiene esta collection.

Para hacerlo desde shell debemos usar la siguiente consulta:

```
db["users"].count()
```

y nos dará el mismo resultado 185

¿Cuántos cines existen en el estado de California?

The screenshot shows the MongoDB Compass interface. The left sidebar displays connections and collections, with 'film\_entertainment.theaters' selected. The main area shows a list of documents from the 'theaters' collection. A search bar at the top contains the query '{ "location.address.state": "CA" }'. Below the search bar are buttons for 'Generate query', 'Explain', 'Reset', 'Find', and 'Options'. The results list shows 169 documents, each representing a theater. Each document includes fields like '\_id', 'theaterId', and 'location'. The first few documents are as follows:

- ObjectID('59a47286cfa9a3a73e51e72e'), theaterId: 1098, location: Object
- ObjectID('59a47286cfa9a3a73e51e734'), theaterId: 1099, location: Object
- ObjectID('59a47286cfa9a3a73e51e737'), theaterId: 1018, location: Object
- ObjectID('59a47286cfa9a3a73e51e739'), theaterId: 1011, location: Object
- ObjectID('59a47286cfa9a3a73e51e73a'), theaterId: 1019, location: Object
- ObjectID('59a47286cfa9a3a73e51e73b'), theaterId: 101, location: Object
- ObjectID('59a47286cfa9a3a73e51e73c'), theaterId: 1021, location: Object

Desde la pestaña documents en el tabulador, ingresamos la consulta, indicando el orden anidado de el dato que queremos filtrar, en este caso el estado de California. Así la consulta es {"location.address.state":"CA"} y el resultado es de 169 cines en el estado de California.

Desde shell la consulta sería:

```
db["theaters"].find({"location.address.state":"CA"}).count()
```

Que es casi igual a la consulta hecha desde la pestaña, pero con la diferencia que le incluimos un count para que nos dé el número de documentos que cumplen con esa condición.

¿Cuál fue el primer usuario en registrarse?

The screenshot shows the MongoDB Compass interface. The left sidebar lists connections and databases, with 'ItAcademy' selected. Under 'ItAcademy', the 'film\_entertainment' database is chosen, and its 'users' collection is active. The main area shows a query builder for the 'users' collection. The 'Find' tab is selected, and the query is defined as follows:

- Project:** { field: 0 }
- Sort:** { '\_id': 1 }
- Collection:** { locale: 'simple' }
- Index Hint:** { field: -1 }

Below the query builder, the results pane displays a single document for Ned Stark:

```
_id: ObjectId('59b99db4cfa9a34dcd7885b6')
name: "Ned Stark"
email: "sean_beansgameofthron.es"
password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crlJ1Vu'
```

Para resolver esta consulta tenemos dos opciones; La primera la utilizamos en el tab find: en opciones, <sort> indicamos la condición { "\_id": 1 } que solicita que nos de el usuario con el id = 1, idealmente debería ser el primer usuario.

En shell, utilizamos la segunda opción que es <createdAt> que ordena por fecha ascendente y <limit(1)> que nos devuelve el primer documento, osea el mas antiguo

```
db["users"].find().sort({ "createdAt": 1 }).limit(1)
_id: ObjectId('59b99db4cfa9a34dcd7885b6'),
name: 'Ned Stark',
email: 'sean_beansgameofthron.es',
password: '$2b$12$UREFwsRUoyF0CRqGNK0Lz00HM/jLhgUCNNIJ9RJAqMUQ74crlJ1Vu'
```

¿Cuántas películas de comedia existen en nuestra base de datos?

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'ITAcademy' as the selected connection, with 'movies' highlighted under the 'film\_entertainment' database. The main area displays the 'movies' collection with 23.5K documents. A search bar at the top has the query '{ genres: "Comedy" }'. Below it, a table shows two document snippets. The first snippet is for a movie with ID '573a1390f29313caabcd4803', which is an animation by Winsor McCay. The second snippet is for another movie with ID '573a1390f29313caabcd50e5', which is about Gertie the Dinosaur.

```
_id: ObjectId("573a1390f29313caabcd4803")
plot: "Cartoon figures announce, via comic strip balloons, that they will move."
genres: Array (3)
  0: "Animation"
  1: "Short"
  2: "Comedy"
runtime: 7
cast: Array (1)
  name: "Comments": 1
poster: "https://m.media-amazon.com/images/M/W5BYzgDNjNHNTctWjUsM10ZWU4LWI3Zj_"
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..."
languages: Array (1)
released: 1911-04-08T00:00:00.000+00:00
directors: Array (2)
writers: Array (2)
awards: Object
lastupdated: "2015-08-29 01:09:03.030000000"
year: 1911
imdb: Object
countries: Array (1)
type: "movie"
tomatoes: Object

_id: ObjectId("573a1390f29313caabcd50e5")
plot: "The cartoonist, Winsor McCay, brings the Dinosaur back to life in th..."
genres: Array (3)
runtime: 12
cast: Array (3)
name: "Comments": 1
poster: "https://m.media-amazon.com/images/M/W5BMTQxNzI4ODQ3NF5BM158anBnXftZT_"
title: "Gertie the Dinosaur"
fullplot: "Winsor Z. McCay bets another cartoonist that he can animate a dinosaur..." languages: Array (1)
```

Para resolver esta consulta utilizamos el tab de <find> y escribimos la consulta {“genres”: “Comedy”} Esto nos devuelve todos los documentos que cumplen con la condición de contener en su género a comedia, y además nos da el total de documentos, que son 7024.

Desde shell utilizamos la siguiente consulta que nos devuelve el conteo de películas que tienen como género “comedia”:

```
db["movies"].find({ "genres": "Comedy" }).count()
7024
```

## Ejercicio 2

Muéstrame todos los documentos de las películas producidas en 1932, pero que el género sea drama o estén en francés

```
_id: ObjectId('573a1391f29313caabed9458')
plot: "A young artist draws a face at a canvas on his easel. Suddenly the mou..."
runtime: 85
rated: "UNRATED"
cast: Array(4)
num_aflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BYWY30DE5ZWEtYJlmY10NJA4LTk4ZWw."
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
writers: Array(1)
awards: Object
year: 1932
imdb: Object
countries: Array(1)
type: "movie"
tomatoes: Object

_id: ObjectId('573a1392f29313caabed9a76')
plot: "In this romance a librarian takes a cruise and falls for an unobtainab..."
genres: Array(2)
runtime: 85
rated: "APPROVED"
cast: Array(5)
num_aflix_comments: 3
poster: "https://m.media-amazon.com/images/M/MV5BMjEzNTUSNTk2N15BM15BanBnKkFtZTw."
title: "Forbidden"
fullplot: "On a cruise to Cuba, Lulu Smith falls in love with Bob Grover. Back ho..."
languages: Array(2)
released: 1932-01-15T00:00:00.000+00:00
directors: Array(1)
writers: Array(2)
```

Para resolver esta consulta en el tab de <find> escribimos el siguiente script:  
{"year": 1932, "\$or": [{"genre": "Drama"}, {"languages": "French"}]}

El operador <\$or> se escribe con el signo <\$> porque es un operador lógico del lenguaje de consultas de Mongodb. Todo operador especial (\$or, \$and, \$gt, \$in, etc) lleva el signo <\$> al principio para diferenciarlo de los nombres normales de los campos.

Y en shell, utilizamos el siguiente script:

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

Obtuvimos 6 documentos.

### Ejercicio 3

Muéstrame todos los documentos de películas estadounidenses que tengan entre 5 y 9 premios que fueron producidas entre 2012 y 2014.

The screenshot shows the MongoDB Compass interface with a connection to 'ItAcademy/film\_entertainment.movies'. The 'movies' collection is selected. A query is run with the following parameters:

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

The results pane displays two document snippets. The first snippet is for the movie 'The Secret Life of Walter Mitty' (ID: 573a13bf29313caabd29366), which has 6 wins and 13 nominations. The second snippet is for the movie 'After Their Cave Is Destroyed' (ID: 573a13bf29313caabd45772), which has 9 wins and 13 nominations.

Para resolver esta consulta utilizamos el siguiente script: {countries: "USA", awards.win: { \$gte: 5, \$lte: 9 }, year: { \$gte: 2012, \$lte: 2014 }}

Igual que el operador `<$or>`, los operadores especiales para mayor o igual que, y menor o igual que son `<$gte>` `<$lte>` respectivamente. hacemos filtros por countries, awards y año. Y el resultado es 166 documentos que cumplen con estas características.

En shell la consulta es:

```
db["movies"].find({ "countries": "USA", "awards.wins": { "$gte": 5, "$lte": 9 }, "year": { "$gte": 2012, "$lte": 2014 } })  
166 documentos
```

## Nivel 2

### Ejercicio 1

Cuenta cuántos comentarios escribe un usuario que utiliza "GAMEOFTHRON.ES" como dominio de correo electrónico.

name	email	movie_id	text	date
Talisa Maegyr	"oona_chaplin@gameofthron.es"	ObjectId('573a1390f29313caabcc41b1')	Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio ture _	1998-08-22T11:45:03.000+00:00
Petyr Baelish	"stannis_gillett@gameofthron.es"	ObjectId('573a1390f29313caabcc421b')	Quo deserunt ipsam ipsum. Tenetur eos remo nam sint praesentium minus _	2001-07-13T19:25:00.000+00:00
Olly	"brenock_o'connor@gameofthron.es"	ObjectId('573a1390f29313caabcc437c')	Persipiciatis sti paratur quas. Perferendis officia harum ipsum deleni_	2005-01-04T13:49:05.000+00:00
Theon Greyjoy	"alicia_allen@gameofthron.es"	ObjectId('573a1390f29313caabcc437c')	Dicta asperiores necessitatibus corporis. Quidem fugiat eius animi fug_	2000-12-06T07:26:20.000+00:00
Jorah Mormont	"laine_glen@gameofthron.es"	ObjectId('573a1390f29313caabcc44d3')	Hinus sequi incident cum magnam. Quam voluptatum vitae ab voluptatum c_	1994-02-18T18:52:31.000+00:00

Para resolver esta consulta utilizamos el siguiente script: { "email": { "\$regex": "gameofthron\\\.es\$", "\$options": "i" } } Donde:

- "\$regex": > busca el patrón que coincide con correos que terminan en gameofthron.es.
- "\\\.es\$": El doble backslash (\\.) es necesario para escapar el punto (.).
- "\$options": "i" > hace la búsqueda insensible a mayúsculas/minúsculas.

Y el resultado es que no hay un solo usuario, sino varios que utilizan ese dominio. Y en total suman 22841.

En shell la consulta sería:

```
db["comments"].countDocuments({email: { $regex: /gameofthron\.es$/i }})  
22841
```

## Ejercicio 2

¿Cuántos cines existen en cada código postal situados dentro del estado Washington DC (DC) ?

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

- Left sidebar (Compass):** Shows connections to ItAcademy, including admin, config, film\_entertainment, theaters, users, and local.
- Top navigation:** Welcome, film\_entertainment, theaters, Aggregations (selected), Schema, Indexes, Validation.
- Query bar:** Cuentos cines existen en cada código postal situados dentro del estado Washington DC (DC).
- Aggregation pipeline:**
  - Stage 1: \$match:** Filters documents where "location.address.state": "DC".
  - Stage 2: \$group:** Groups by "location.address.zipcode" and calculates "total\_theaters" (sum of 1).
- Output preview:** Shows three sample documents from the first stage and three sample documents from the second stage.

Para resolver esta consulta fuera de shell, tenemos que usar la pestaña aggregations, que es la que nos permite agrupar y contar por grupos. Utilizamos el siguiente pipeline: [{ \$match: { "location.address.state": "DC" } }, { \$group: { \_id: "\$location.address.zipcode", total\_cines: { \$sum: 1 } } }]. Donde:

- \$match: filtra los documentos cuyo campo state sea "DC".
- \$group: agrupa por el código postal (zipcode) y cuenta cuántos cines hay en cada uno.

y los resultados son:

The screenshot shows the MongoDB Shell output for the aggregation query:

```
ALL RESULTS
Showing 1-3 count results < > ▾ ⌂ ⌂
```

_id	total_theaters
"20016"	1
"20010"	1
"20002"	1

Para hacerlo desde Shell utilizamos el siguiente script

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

### Nivel 3

#### Ejercicio 1

Encuentra todas las películas dirigidas por John Landis con una puntuación IMDb (Internet Movie Database) de entre 7,5 y 8.

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar is open, showing a connection named 'ItAcademy' with several databases listed: admin, config, film\_entertainment, local, sessions, theaters, and users. The 'film\_entertainment' database is selected. In the main area, the 'movies' collection is selected under 'Documents'. A search query is entered in the search bar: { "directors": "John Landis", "imdb.rating": { "\$gte": 7.5, "\$lte": 8 } }. Below the search bar, there are buttons for 'Generate query', 'Explain', 'Reset', 'Find', and 'Options'. The results pane shows two documents matching the query. The first document is for the movie 'Faber College' (id: 573a1397f29313caabce6d94), and the second is for 'The Blues Brothers' (id: 573a1397f29313caabce76f7).

```
_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 student body for their spring break party. He enlists the help of his former student, John Goodman, to track them down and bring them back to school. The students, led by John Goodman's son, are forced to travel across the country to get home before Wormer can catch up to them."
  genres: Array (1)
    rated: "R"
    runtime: 82
  title: "Fraternal House"
  lastupdated: "2015-09-13 00:02:47.883000000"
  languages: Array (2)
  writers: Array (3)
  type: "movie"
  tomatoes: Object
    poster: "https://m.media-amazon.com/images/M/MV5BM2M2ZDA4MTYtOGJjM100Tg5LWI1ZT.."
    numReviews: 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 the town from a group of bullies who have taken over the town. The band includes Elwood, the bluesy piano player; Jake, the lead singer; and the bluesy guitar player, Little Edie. They play their hearts out and win the town over, but Jake is captured again and sent to jail. The townspeople rally around him and release him, and he and Elwood go on to become big stars."...
```

Para resolver esta consulta utilizamos en el tab de find, el siguiente script:

```
{"directors": "John Landis",
```

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

Donde básicamente señalamos dentro de "directors" que director queremos, y en "imdb.rating" que tenga un raiting de igual o mayor a 7.5 o igual o menor a 8.

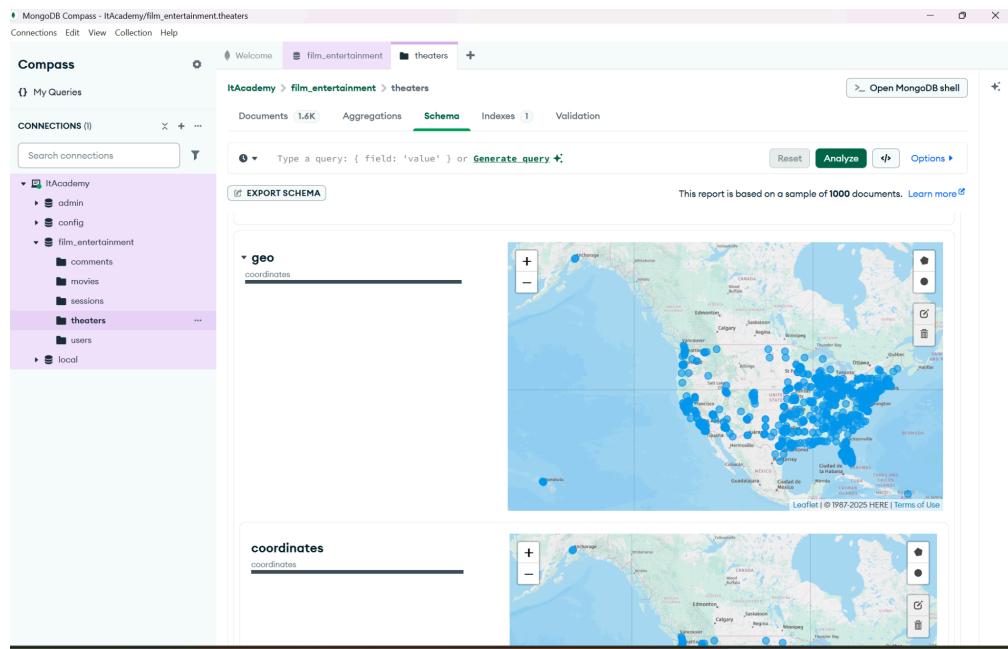
y en shell utilizamos el siguiente script:

```
db.movies.find({directors: "John Landis", "imdb.rating": { $gte: 7.5, $lte: 8 } }, {title: 1, year: 1, "imdb.rating": 1, _id: 0})
```

Donde a más de hacer la consulta principal, indicamos cuáles son los campos que queremos que nos devuelva de los documentos que coincidan con estas condiciones, para facilitar la visualización.

## Ejercicio 2

Muestra en un mapa la ubicación de todos los teatros de la base de datos.



Para poder acceder al mapa de los cines, nos tenemos que desplazar a la pestaña <Schema> desde la collection theater. Una vez ahí damos clic a <Analize Schema> nos desplazamos hacia abajo y en los desplegables le damos clic a <Location> luego a <Geo> y listo tenemos el mapa con los cines.