

Nivell 1

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

Bajo Compas de Mongo, crearemos una base de datos “sprint_09” en nuestro caso.

Luego en cada collection (tabla) importaremos los datos proporcionados.

- Exercici 1

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

Para buscar objetos en MongoDB, usamos el método .find() es el homologo de un Select en SQL.

Luego al igual que SQL usamos limit, en este caso.limit(2) para solo mostrar los dos primeros resultados.

```
db.comments.find().limit(2)
{
  _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 voluptatum voluptates neque culpa molestias. Voluptate unde nulla temporibus ullam.',
  date: 2012-03-26T23:20:16.000Z
}
{
  _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. Commodi nisi sit placeat rerum vero cupiditate neque. Dolorum nihil vero animi.',
  date: 1987-02-10T00:29:36.000Z
}
```

- Quants usuaris tenim registrats?

Para contar elementos en Mongo debemos usar `.countDocuments()`. Especificamos la tabla a contar para que nos muestre los resultados, en este caso `185`.

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

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

Ahora con `countDocument` filtramos, en este caso tenemos que entrar dentro de donde se guarda la localización (`location`) y poner la ruta hasta el estado "`location.address.state`" en este caso especificamos con las iniciales de California "`CA`".

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

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

Debido a como se almacenan los datos en mongo, estos crean los ids por orden de llegada, así que nos interesaría filtrarlo por ID y establecer un limit de 1 para mostrar únicamente el primer dato.

En esta caso es "Ned Stark"

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

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

De la misma manera volvemos a usar `countDocuments`, buscamos donde se almacenan las categorías de las películas en este caso "genres" y además aseguramos mostrar únicamente películas especificando el type"

Quedaría entonces "type:"movie",genres:"Comedy"

```
> db.movies.countDocuments({ type: "movie", genres: "Comedy" })  
< 7002
```

Además, si solo nos gustaría contar aquellas películas categorizadas como Comedy sin ningún otro tipo de género deberemos filtrar únicamente por la primera entrada de "genres" esto lo hacemos especificando:

```
> db.movies.countDocuments({ type: "movie", genres: { $size: 1, $elemMatch: { $eq: "Comedy" } } })  
< 1144
```

- Exercici 2

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

Volvemos a usar un find especificando que buscamos películas del año 1932 pero en este caso como nos pide un condicional, en mongo es "\$or" y especificamos la siguiente condición.

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

Estos serían los resultados:

```
{
  _id: ObjectId('573a1391f29313caabcd9458'),
  plot: 'A young artist draws a face at a canvas on his easel. Suddenly the mouth on the drawing comes into life and starts talking. The artist tries to wipe i
  runtime: 55,
  rated: 'UNRATED',
  cast: [
    'Enrique Rivero',
    'Elizabeth Lee Miller',
    'Pauline Carton',
    'Odette Talazac'
  ],
  num_mflix_comments: 1,
  poster: 'https://m.media-amazon.com/images/M/MV5BYWY3ODE5ZWEtYjlmYi00NjA4LTk4ZWYtMzBhZDE5MjY0YTYxXkEyXkFqcGdeQXVyNzI4MDMyMTU@._V1_SY1000_SX677_AL_.jpg',
  title: 'The Blood of a Poet',
  lastupdated: '2015-09-16 13:13:05.537000000',
  languages: [
    'French'
  ],
  released: 2010-05-20T00:00:00.000Z,
  directors: [
```

- 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 establecer un parámetro de rango de búsquedas, lo hacemos especificando la primera condición con \$gte y acabando con la ultima \$lte. Ambos incluidos. Nos daría 162 películas

```
> db.movies.find({
  "type": "movie",
  "countries": "USA",
  year: { $gte: 2012, $lte: 2014 },
  "awards.wins": { $gte: 5, $lte: 9 }
})
< {
  _id: ObjectId('573a13acf29313caabd29366'),
  fullplot: "The manager of the negative assets sector of Life magazine, Walter Mitty, has been working for sixteen years for the magazine and has a tedious life, not going any",
  imdb: {
    rating: 7.4,
    votes: 211230,
    id: 359950
  },
  year: 2013,
  plot: "When his job along with that of his co-worker are threatened, Walter takes action in the real world embarking on a global journey that turns into an adventure more ext",
  genres: [
    'Adventure',
    'Comedy',
    'Drama'
  ],
}
```

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

Nivell 2

- Exercici 1

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

Para buscar en las cadenas de texto necesitamos usar la función regex el \$ nos permite especificar la localización, es decir al final.

Acabamos haciendo un count a esta búsqueda para mostrar todos los comentarios hechos por los usuarios con el dominio.

```
> db.comments.find({  
  email: {$regex: /@gameofthron.es$/i}  
}).count()  
< 22841
```

- Exercici 2

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

Ahora necesitaremos “aggregate” para poder establecer una condición y calcular un dato.

En este caso la condición es encontrar todos aquellos teatros que se encuentren en DC
“location.address.state” seria la ruta donde se almacena este dato.

Agruparlos por zipcode “location.address.zipcode” es donde se almacena, y contar 1 por cada uno que encuentre.

```
db.theaters.aggregate([
  { $match: { "location.address.state": "DC" } },
  { $group: { _id: "$location.address.zipcode", count: { $sum: 1 } } },
  { $sort: { count: -1 } }
])
```

```
< {
  _id: '20010',
  count: 1
}
{
  _id: '20016',
  count: 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.

Al igual que hemos hecho con anterioridad, find() al autor y añadiendo la ruta a el imdb rating filtrado.

```
> db.movies.find({
  directors: "John Landis",
  "imdb.rating": { $gte: 7.5, $lte: 8 }
})
{
  _id: ObjectId('573a1397f29313caabce6d94'),
  fullplot: "Faber College has one frat house so disreputable it will take anyone. It has a second one full of white, anglo-saxon, rich young men who are so sa
  imdb: {
    rating: 7.6,
    votes: 84834,
    id: 77975
```


- Exercici 2

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

Para generar el mapa, debemos ir al apartado de Schema sobre “Theatres” y analizarlo, en el apartado de location nos aparecerá el mapa. Este únicamente nos mostrará 1000 resultados.

