

# Nivell 1

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

## - Exercici 1

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

Seleccionamos la colección “ comments” y vamos a aggregations.

Allí en stage ponemos limit = 2

The screenshot shows the MongoDB Aggregations interface. At the top, there are tabs for Documents (50.3K), Aggregations (selected), Schema, Indexes (1), and Validation. Below the tabs, there's a section for the aggregation pipeline. The first stage is named '\$limit' and has a value of 2. The output of this stage is shown as a sample of 2 documents. The first document is for Andrea Le, and the second is for Greg Powell. The interface also includes buttons for 'Generate aggregation', 'Explain', 'Export', 'Run', and 'Options'. At the bottom, there are buttons for 'SAVE', 'CREATE NEW', 'EXPORT TO LANGUAGE', 'PREVIEW', 'STAGES', 'TEXT', 'WIZARD', and a settings icon.

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

Output after \$limit stage (Sample of 2 documents)

<pre>{   "_id": ObjectId("5a9427648b0beeb69579cc"),   "name": "Andrea Le",   "email": "andrea_le@fakegmail.com",   "movie_id": ObjectId("573a1390f29313caabcd418c"),   "text": "Rem officiiis eaque repellendus amet eos doloribus. Porro dolor volupta...",   "date": 2012-03-26T23:20:16.000+00:00 }</pre>	<pre>{   "_id": ObjectId("5a9427648b0beeb69579cc"),   "name": "Greg Powell",   "email": "greg_powell@fakegmail.com",   "movie_id": ObjectId("573a1390f29313caabcd418c"),   "text": "Tenetur dolorum molestiae praesentium unde quod por...",   "date": 1987-02-10T00:29:36.000+00:00 }</pre>
--	--

Vemos que corresponden a los comentarios de Andrea Le y a Greg Powell

- Quants usuaris tenim registrats?

Vamos al apartado “Aggregations” y le decimos que cuente el número de documentos que hay en la colección de Usuarios.

Documents 185 Aggregations Schema Indexes 1 Validation

Count Generate aggregation Explain Export Run Options

Ejercicio 2 \_ni... SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage1 Count

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

Output after Count stage (Sample of 1 document)

```
string : 185
```

Obtenemos que hay 185 registros.

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

Usaremos la colección "Theaters" y vamos a la pestaña "Aggregations"

## Stage 1

Aplicaremos un filtro en \$match para indicar que el "location.address.state" ha de ser CA que es la abreviación del estado de California.

```
1 /**
2  * query: The query in MQL.
3  */
4  {
5    "location.address.state": "CA"
6  }
```

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

```
{
  "_id": ObjectId('59a47286cfa9a3a73e51e72e'),
  "theaterId": 1008,
  "location": Object
}
```

```
{
  "_id": ObjectId('59a47286cfa9a3a73e51e72e'),
  "theaterId": 1009,
  "location": Object
}
```

## Stage 2

Usamos el método \$count para contar el número de registros.

Stage2 Count

```
1 /**
2  * Provide the field name for the count.
3  */
4  'Total cines en Canadá'
```

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

```
Total cines en Canadá : 169
```

Sale que en el estado de Canadá hay un total de 169 cines.

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

Usaremos la colección users para llevar a cabo este ejercicio. No hay ningún campo que contenga la fecha de registro, así que usaremos el campo "\_id" para ordenar la colección.

## Stage 1

Con el método \$sort, ordenamos por el campo “\_id”

Stage 1 \$sort

```
1 /**
2  * Provide any number of field/order pair
3  */
4 {
5   "_id" : 1
6 }
```

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

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

## Stage 2

Con el método \$limit, indicamos que sólo queremos el primer registro.

Stage 2 \$limit

```
1 /**
2  * Provide the number of documents to limit
3  */
4 {
5   "_id" : 1,
6   "limit" : 1
7 }
```

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

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

Obtenemos que Ned Stark fue el primer usuario que se registró en la colección.

## • Cuantes películas de comedia hi ha en la nostra base de dades?

Para esta consulta usaremos la colección “Movies”.

## Stage 1

Con el método \$match, seleccionamos el género “Comedy” del campo “Genres”.

Stage 1 \$match

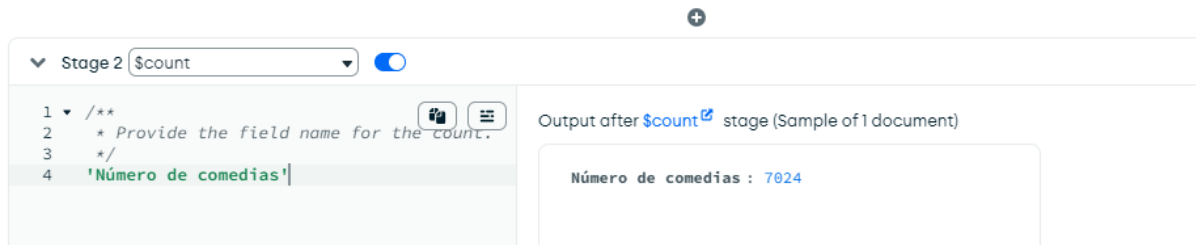
```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "genres" : "Comedy"
6 }
```

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

```
{ "_id": ObjectId('573a1390f29313caab4803'),
  "plot": "Cartoon figures announce, via comic strip balloons, that they will mov...",
  "genres": ["Comedy"],
  "runtime": 7,
  "cast": ["Cartoon figures"],
  "num_mflix_comments": 1,
  "poster": "https://m.media-amazon.com/images/M/MV5BYzg2NjNhNTctM..." }
```

## Stage 2

Con el método \$count, contaremos el número de registros.



Obtenemos que en la colección hay 7024 películas catalogadas como comedia.

## - 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 esta consulta, usaremos la colección “movies”.

Stage1:

Aquí se han de cumplir dos condiciones: Que sean producidas en 1932 y sean películas, por eso usaremos el operador \$and.

Para ello usaremos los campos “year” y “type”. En type hay dos opciones que son películas o series.

```
$and:[{"year":1932}, {"type":"movie"}],
```

Por otra parte, han de ser o bien en francés o dramas. Para eso usaremos los campos languages y genres anidadas en el operador \$or.

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

Me resulta curioso ver que hay películas que no tiene el campo languages. Por ejemplo, los objetos 573a1390f29313caabcd548c y 573a1390f29313caabcd4135.

Consulta final en la pestaña “Aggregations”

```
{ $and:[{"year":1932}, {"type":"movie"}],
```

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

Stage 1 \$match

```
1 /**
2  * query: The query in MQL.
3  */
4
5  ({
6    $and:[{"year":1932}, {"type":"movie"}],
7    $or:[{"genres":"Drama"}, {"languages":"French"}]
8  })
9
10
```

Output after \$match stage (Sample of 10 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/MV5BYWY3ODE5ZWEtY...",
  ...
}
```

```
{
  "_id": ObjectId('573a1392f29313caa...'),
  "plot": "Junta is hated by the pe... village where she lives, i...",
  "genres": Array (3),
  "runtime": 85,
  "cast": Array (4),
  "poster": "https://m.media-amazon.com/images/M/MV5I...",
  "title": "The Blue Light"
}
```

## Stage 2

Usando el método \$count, calcularemos el número de películas que cumplen con estas condiciones.

Stage 2 \$count

```
1
2  write the field name for the count.
3
4  :ulas rodadas en 1932 que sean dramas o estén en francés
```

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

```
{
  "Películas rodadas en 1932 que sean dramas o estén en francés": 18
}
```

En total nos salen 18 películas producidas en 1932 que sean drama o estén en francés.

## - 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 esta consulta usaremos la colección "movies".

Condiciones:

1 - Peliculas de EEUU.

```
"countries" = "USA"
```

2 – Que tengan más de 5 premios y menos que 9

```
"awards.wins" : {$gte:5,$lte:9}
```

Gte = greater or equal to

Lte = less or equal to

3 – Producidas entre 2012 y 2014

```
"year":{$gte:2012,$lte:2014}
```

4 – Que sean películas

```
"type": "movie"
```

### Stage 1

Mediante el método \$match, incluiremos todas las condiciones del enunciado del ejercicio arriba detallado.

The screenshot shows the MongoDB Compass interface for Stage 1. The query editor on the left contains a \$match query: 

```
1 /**  
2  * query: The query in MQL.  
3  */  
4 {  
5    "countries": "USA",  
6    "awards.wins" : {$gte:5,$lte:9},  
7    "year":{$gte:2012,$lte:2014},  
8    "type": "movie"  
9  }  
10 }
```

 The right panel, titled 'Output after \$match stage (Sample of 10 documents)', displays two sample documents. The first document is for 'The manager of the negative assets sector of Life magazine, Walter Mit...' and the second is for 'After their cave is destroyed, a caveman family must trek...'.

### Stage 2

Mediante el método \$count, contaremos cuantas películas cumplen con todos los criterios.

The screenshot shows the MongoDB Compass interface for Stage 2. The query editor on the left contains a \$count query: 

```
1 /**  
2  * Provide the field name for the count.  
3  */  
4 'Películas Estadounidense, con más de cin'
```

 The right panel, titled 'Output after \$count stage (Sample of 1 document)', displays a single document: 

```
Películas Estadounidense, con : 162  
más de cinco premios pero  
menos de 9 y producidas entre  
2012 y 2014
```

Hay un total de 162 películas estadounidense producidas entre 2012 y 2014 y que tiene más de cinco premios, pero menos que 9.

## Nivell 2

### - Exercici 1

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

Para resolver este ejercicio vamos a utilizar la función \$regex la cual nos ayuda a identificar patrones dentro de strings. En este caso necesitamos que el mail acabe en "gameofthron.es". Con lo cual utilizaremos la expresión \$ al final.

\$regex: /gameofthron.es\$/

Para llevar a cabo esta consulta usaremos la colección "comments", en la pestaña "Aggregations"

#### Stage 1

Con el método \$match, buscaremos los mails que acaben en "gameofthron.es"

"email" : {\$regex: /gameofthron.es\$/}

#### Stage 2

The screenshot shows the MongoDB Aggregations Builder interface. Stage 1 is configured with a \$match aggregation using the regex /gameofthron.es\$/ on the email field. Stage 2 is currently empty. The output of Stage 1 shows sample documents with fields like \_id, name, email, movie\_id, text, and date.

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   email: {
6     $regex: /gameofthron.es$/
7   }
8 }
```

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

Document 1	Document 2
<pre>{   "_id": ObjectId("5a9427648b0beeb69579d0"),   "name": "Talisa Maegyr",   "email": "oona_chaplin@gameofthron.es",   "movie_id": ObjectId("573a1390f29313caabed41b1"),   "text": "Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..." }</pre>	<pre>{   "_id": ObjectId("5a9427648b0beeb69579d0"),   "name": "Petyr Baelish",   "email": "aidan_gillen@gameofthron.es",   "movie_id": ObjectId("573a1390f29313caabed41b1"),   "text": "Quo deserunt ipsam ipsum nemo nam sint praesentium ..." }</pre>

Con el método \$count, contaremos los casos que hay.

The screenshot shows the MongoDB Aggregations Builder interface. Stage 2 is configured with a \$count aggregation. The output shows the total count of documents matching the criteria in Stage 1.

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

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

```
{
  "Comentarios": 22841
}
```

La consulta nos devuelve que el usuario con dominio @gameofthron.es ha escrito un total de 22841 comentarios.

## - Exercici 2

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

Aquí tindrèm que filtrar los estats para quedarnos con los cines de Washington DC y luego agrupar por código postal y contar cuántos cines hay en cada código postal.

Esta consulta la haremos sobre la colección "theaters"

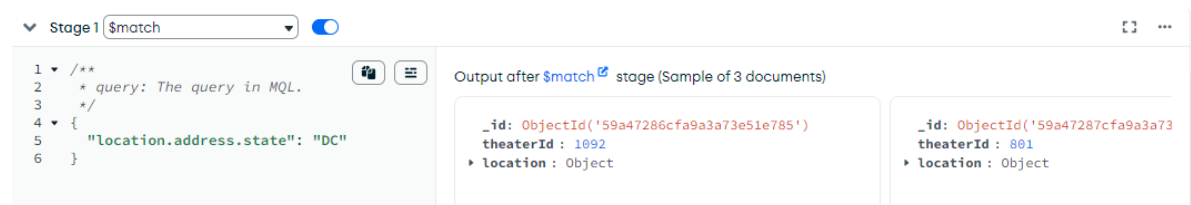
Según Google, Washington DC se abrevia D.C que significa "District of Columbia".

El filtro será el siguiente: "location.address.state" : "DC"

Entramos en la pestaña "aggregations"

Stage 1

Usamos \$match para filtrar el estado DC.



Stage 1 \$match

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   "location.address.state": "DC"
6 }
```

Output after \$match stage (Sample of 3 documents)

```
{
  "_id": ObjectId('59a47286cfa9a3a73e51e785'),
  "theaterId": 1092,
  "location": Object
}
```

```
{
  "_id": ObjectId('59a47287cfa9a3a73e51e786'),
  "theaterId": 801,
  "location": Object
}
```

Stage 2.

Procedemos a la agrupación con el método \$group y a contar casos.



```
1 [
2   {
3     $match:
4       /**
5        * query: The query in MQL.
6        */
7       {
8         "location.address.state": "DC"
9       }
10  },
11  {
12    $group:
13      /**
14       * _id: The id of the group.
15       * fieldN: The first field name.
16       */
17      {
18        _id: "$location.address.zipcode",
19        Num_cines: {
20          $count: {}
21        }
22      }
23  }
24 ]
```

PIPELINE OUTPUT

Sample of 3 documents

```
{
  "_id": "20016",
  "Num_cines": 1
}
```

```
{
  "_id": "20002",
  "Num_cines": 1
}
```

```
{
  "_id": "20010",
  "Num_cines": 1
}
```

La consulta arroja que sólo hay tres códigos postales en Washington DC y que en cada código postal sólo hay un cine.

## Nivell 3



## - Exercici 1

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

Para esta consulta utilizaremos la colección "movies"

Filtraremos por:

1 – Que sean películas y no series.

```
"type": "movie"
```

2 – Que estén dirigidas por John Landis

```
"directors": "John Landis"
```

3 - puntuació IMDb (Internet Movie Data base) d'entre 7,5 i 8.

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

Gte = greater or equal than , lte = less or equal than

Ahora vamos a la pestaña "Aggregations" y añadimos estas condiciones con el método \$match en el stage 1.

The screenshot shows the MongoDB Compass interface. On the left, the query editor displays the following query:

```
1 //**
2 * query: The query in MQL.
3 */
4 {
5   "type": "movie",
6   "directors": "John Landis",
7   "imdb.rating": {'$gte': 7.5, '$lte': 8}
8 }
```

On the right, the 'Output after \$match stage (Sample of 4 documents)' section shows two sample documents:

```
{
  "_id": ObjectId('573a1397f29313caabce6d94'),
  "fullplot": "Faber College has one frat house so disreputable it will take anyone. ...",
  "imdb": {
    "year": 1978,
    "plot": "At a 1962 college, Dean Vernon Wormer is determined to expel the entire...",
    "genres": Array (1),
    "rated": "R"
  }
}
```

```
{
  "_id": ObjectId('573a1397f29313caa...'),
  "plot": "Jake Blues, just out from together his old band to : ...",
  "genres": Array (3),
  "runtime": 133,
  "rated": "R",
  "cast": Array (4),
  "num_mflix_comments": 1,
  "poster": "https://m.media-..."
}
```

La consulta arroja 4 resultados, pero en la pestaña "Aggregations" se ve muy mal. Repetimos la consulta en la pestaña "documents".



+++++++ unquote ++++++

....