

# TASK S5.01. Queries with MongoDB

## LEVEL 1:

-Create a database with MongoDB using the attached files as collections.

The screenshot shows the Compass MongoDB interface. On the left, the connections sidebar lists 'mavi' and 'sprint5'. Under 'sprint5', there are five collections: 'comments', 'movies', 'sessions', 'theaters', and 'users'. Each collection card displays its storage size (4.10 kB), document count (e.g., 50 K for comments), average document size (e.g., 284.00 B for comments), index count (1), and total index size (4.10 kB). A message at the bottom indicates 'Import completed. 185 documents imported.'

## Exercise 1:

-Show the first 2 comments in the database.

The screenshot shows the Compass MongoDB interface with the 'comments' collection selected under the 'mavi > sprint5' connection. The query parameters are set to sort by date (date:1) and limit the results to 2. The results table shows two documents. The first document has an \_id of 5a9427648b0beeb695e67e, a name of 'Mercedes Tyler', an email of 'mercedes\_tyler@fakegostail.com', a movie\_id of 573a1344f29313cabcd117df, a text field containing a long string of Latin text, and a date of 2017-01-01T01:07:09.000+00:00. The second document has an \_id of 5a942765ab0beeb695e6fe, a name of 'Jon Snow', an email of 'kit\_harington@gamofthrone.es', a movie\_id of 573a1344f29313cabcd117df, a text field containing another long string of Latin text, and a date of 2017-01-01T09:37:49.000+00:00.

Compass

CONNECTIONS (1) + ...

Search connections

- movi
  - admin
  - config
  - local
  - sprint5
    - comments
    - movies
    - sessions
    - theaters
    - users

\_MONGOSH

```
> use sprint5
< switched to db sprint5
> db.comments.find().sort({date:1}).limit(2)
< [
  {
    _id: ObjectId('5a9427640b0beeb695e67e'),
    name: 'Mercedes Tyler',
    email: 'mercedes_tyler@fakegmail.com',
    movie_id: ObjectId('573a13a4f29313caabd117df'),
    text: 'Optio totam dolores magni. Enim ratione fuga tempora voluptatum est cumque animi. Quia nam doloribus corrupti id voluptate esse. Tempore maiores omnis aliquam ad quisquam qui date: 1970-01-01T01:07:09.000Z
  }
  {
    _id: ObjectId('5a9427650b0beeb6969efc'),
    name: 'Jon Snow',
    email: 'kit_harington@gameofthron.es',
    movie_id: ObjectId('573a13a4f29313caabd117df'),
    text: 'Dolorum animi tempora ullam quas. Iusto nobis reprehenderit aspernatur cupiditate a tempore. Commodi alias expedita dolore explicabo ipsam.',
    date: 1970-01-01T09:37:49.000Z
  }
]
sprint5|
```

-How many users are registered?

Compass

CONNECTIONS (1) + ...

Search connections

- movi
  - admin
  - config
  - local
  - sprint5
    - comments
    - movies
    - sessions
    - theaters
    - users

movi > sprint5 > users

Documents: 185 Aggregations Schema Indexes 1 Validation

\$count

Generate aggregation + Explain Export Run Options ▾

PREVIEW STAGES TEXT WIZARD

userCount SAVE + CREATE NEW EXPORT TO LANGUAGE

name : "Ned Stark"
email : "sean\_bean@gameofthron.es"
password : "\$2b\$12\$UREFwsRUoyF0CRqGNK8Lz0H..

name : "Robert Baratheon"
email : "mark\_addy@gameofthron.es"
password : "\$2b\$12\$yGqxLG9LzPxA2xV0huPnSOzd..

name : "Jaime Lannister"
email : "nikolaj\_coster-waldau@gameofthron.es"
password : "\$2b\$12\$6v27wiw0.Ei5Rilvq1zUc./9..

Stage1 \$count

1 /\*+
2 \* Provide the field name for the count.
3 \*/
4 "userCount"

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

userCount : 185

Compass

CONNECTIONS (1) + ...

Search connections

- movi
  - admin
  - config
  - local
  - sprint5
    - comments
    - movies
    - sessions
    - theaters
    - users

\_MONGOSH

```
> use sprint5
< switched to db sprint5
> db["users"].countDocuments()
< 185
sprint5|
```

-How many movie theaters are there in the state of California?

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

- Top Bar:** Welcome, mavi, comments, sprint5, comments, movies, sessions, theaters, users.
- Left Sidebar (Connections):** mavi, admin, config, local, sprint5, comments, movies, sessions, theaters, users.
- Current Collection:** theaters
- Aggregation Pipeline:**
  - Stage 1 (\$match):** Shows the query: `1 //** 2 * query: The query in MQL. 3 */ 4 { 5 "location.address.state": "CA" }`. Output after \$match stage (Sample of 10 documents):
 

<code>_id: ObjectId('59a47286cfa9a3a73e51e72e')</code>	<code>theaterId : 1088</code>	<code>location : Object</code>
<code>_id: ObjectId('59a47286cfa9a3a73e51e734')</code>	<code>theaterId : 1089</code>	<code>location : Object</code>
  - Stage 2 (\$count):** Shows the query: `1 /** 2 * Provide the field name for the count. 3 */ 4 'theaterCount'|`. Output after \$count stage (Sample of 1 document):
 

<code>theaterCount : 169</code>
---------------------------------
- Bottom Panel (MongoSH):**

```
>_MONGOSH
>use sprint5
< switched to db sprint5
>db.theaters.aggregate([
  { $match: { "location.address.state": "CA" } },
  { $count: "total_theaters" }
])
< {
  total_theaters: 169
}
>sprint5>|
```

-Who was the first user to register?

MongoDB Compass - mavi/sprint5.users

Documents 186 Aggregations Schema Indexes 1 Validation

Search connect... \$addFields \$sort \$limit

Generate aggregation Explain Export Run Options

PREVIEW STAGES TEXT WIZARD

**Stage 1: \$addFields**

```

1 /**
2  * newField: The new field name.
3  * expression: The new field expression.
4 */
5 {
6   "createDate": { "$toDate": "$_id" }
7 }
  
```

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

```

_id: ObjectId('59b99db4cf9a34cd7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$URFwsRlOyFOCQnK8Lz0HHM_
createDate: 2017-09-13T21:05:56.000+00:00

_id: ObjectId('59b99db4cf9a34cd7885b7')
name: "Robert Baratheon"
email: "mark_addy@gameofthron.es"
password: "$2b$12$yGoL9LzpxA2vVDhuPnS0Zd...
createDate: 2017-09-13T21:05:56.000+00:00

_id: ObjectId('59b99db4cf9a34cd7885b8')
name: "Daenerys Targaryen"
email: "michelle_fairley@gameofthron.es"
password: "$2b$12$gflaH5Sh1XKNF2i/FTErW6j...
createDate: 2017-09-13T21:05:56.000+00:00
  
```

**Stage 2: \$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('59b99db4cf9a34cd7885b7')
name: "Catelyn Stark"
email: "michelle_fairley@gameofthron.es"
password: "$2b$12$gflaH5Sh1XKNF2i/FTErW6j...
createDate: 2017-09-13T21:05:57.000+00:00

_id: ObjectId('59b99db4cf9a34cd7885b9')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$URFwsRlOyFOCQnK8Lz0HHM...
createDate: 2017-09-13T21:05:56.000+00:00

_id: ObjectId('59b99db4cf9a34cd7885bd')
name: "Petyr Baelish"
email: "aidan_gillen@gameofthron.es"
password: "$2b$12$9qMqYvnikeyYYYY7phK30icb...
createDate: 2017-09-13T21:05:59.000+00:00
  
```

**Stage 3: \$limit**

```

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

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

```

_id: ObjectId('59b99db4cf9a34cd7885b6')
name: "Ned Stark"
email: "sean_bean@gameofthron.es"
password: "$2b$12$URFwsRlOyFOCQnK8Lz0HHM...
createDate: 2017-09-13T21:05:56.000+00:00
  
```

-How many comedy movies are in our database?

Compass

My Queries

Connections (1)

Search connections

mavi comments sprint5 comments movies sessions theaters users

Open MongoDB shell

Documents 23.5K Aggregations Schema Indexes 1 Validation

\$match \$count

Generate aggregation Explain Export Run Options

PREVIEW STAGES TEXT WIZARD

**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('573a1390f29313caabcd4803')
plot: "Cartoon figures announce, via comic strip balloons, that they will mov...
genres: Array (3)
0: "Animation"
1: "Short"
2: "Comedy"
runtime: 7
cast: Array (1)
  
```

**Stage 2: \$count**

```

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

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

```

moviesComedy: 7024
  
```

```

>_MONGOSH
> use sprint5
< switched to db sprint5
> db['movies'].countDocuments({genres:"Comedy"})
< 7824

```

**Exercise 2:** Show me all the documents for films produced in 1932, but the genre is drama, or they are in French.

Aggregations

```

1932drama

```

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

```

_id: ObjectId('573a1391f29313caab9458')
plot: "A young artist draws a face at a
mos...n his easel. Suddenly the
villag...
runtime: 55
rated: "UNRATED"
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media...

```

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

```

title: "The Blood of a Poet"
languages: Array (1)
0: "French"
year: 1932

```

Stages

```

npass
Welcome mavi comments mongosh:mavi sprint5 comments movies mongosh:mavi sessions theaters users +
Queries
ECTIONS (1) × + ...
Arch connect
mavi admin config local sprint5 comments movies sessions theaters users
>_MONGOSH
> use sprint5
< switched to db sprint5
> db.movies.find({
  year: 1932,
  $or: [
    {
      genres: "Drama"
    },
    {
      languages: "French"
    }
  ],
  _id: 0,
  title: 1,
  year: 1,
  genres: 1,
  languages: 1
})
< [
  {
    title: 'The Blood of a Poet',
    languages: [
      'French'
    ],
    year: 1932
  }
  {
    genres: [
      'Drama',
      'Fantasy',
      'Mystery'
    ],
    ...
  }
]
> db.movies.countDocuments({
  year: 1932,
  $or: [
    {
      genres: "Drama"
    },
    {
      languages: "French"
    }
  ],
  _id: 0,
  title: 1,
  year: 1,
  genres: 1,
  languages: 1
})
< 18
sprint5>

```

**Exercise 3:** Show me all documents of American films with between 5 and 9 awards that were produced between 2012 and 2014.

**Compass**

**movi > sprint5 > movies**

**Aggregations**

**USAwards**

**Stage 1 \$match**

```

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

```

**Stage 2 \$project**

```

1 /**
2 * specifications: The fields to
3 * include or exclude.
4 */
5 {
6   "_id": 0,
7   "title": 1,
8   "year": 1,
9   "countries": 1,
10  "awards": 1
11 }

```

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

**Output after \$project stage (Sample of 10 documents)**

## LEVEL 2:

**Exercise 1:** Count how many comments a user writes who uses "GAMEOFTHRON.ES" as their email domain.

**Compass**

**movi > sprint5 > comments**

**Aggregations**

**Untitled - modified**

**Stage 1 \$match**

```

1 /**
2 * query: The query in MQL.
3 */
4 {
5   "email": { $regex: /@gameofthron\.es$/i }
6 }

```

**Stage 2 \$count**

```

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

```

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

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

## Exercise 2: How many movie theaters are there in each zip code located within the state of Washington D.C.(DC)?

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

- Left Sidebar (Connections):** Shows a tree view of databases: mavi, admin, config, local, sprint5, comments, movies, sessions, theaters, users.
- Top Bar:** Welcome, mavi, comments, sprint5, comments, movies, sessions, theaters, users, +, Open MongoDB shell.
- Current Collection:** mavi > sprint5 > theaters
- Aggregation Tab:** Documents (1.6K), Aggregations (selected), Schema, Indexes (1), Validation.
- Search Bar:** \$match, \$group.
- Buttons:** DC, SAVE, CREATE NEW, EXPORT TO LANGUAGE, Generate aggregation, Explain, Export, Run, Options.
- Stages:**
  - 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('59a47286cf9a3a73e51e785')`  
`theaterId: 1092`  
location: Object  
address: Object  
city: "Washington"  
state: "DC"  
zipcode: "20010"  
geo: Object
    - `_id: ObjectId('59a47287cf9a3a73e51ec33')`  
`theaterId: 801`  
location: Object  
address: Object  
city: "Washington"  
state: "DC"  
zipcode: "20016"  
geo: Object
    - `_id: ObjectId('59a47287cf9a3a73e51ec33')`  
`theaterId: 8500`  
location: Object  
address: Object  
street1: "100 Wisconsin Ave Nw"  
city: "Washington"  
state: "DC"  
zipcode: "20016"  
geo: Object  - Stage 2: \$group**

```

1 /**
2 * _id: The id of the group.
3 * field: The first field name.
4 */
5 {
6     "_id": "$location.address.zipcode",
7     "theater_count": { $sum: 1 }
8 }
    
```

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

    - `_id: "20016"`  
`theater_count: 1`
    - `_id: "20016"`  
`theater_count: 1`
    - `_id: "20016"`  
`theater_count: 1`

## LEVEL 3:

**Exercise 1:** Find all films directed by John Landis with an IMDb (Internet Movie Database) rating between 7.5 and 8.

**Compass**

movi > sprint5 > movies

Documents 23.5K Aggregations Schema Indexes Validation

\$match \$project

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE

Generate aggregation Explain Export Run Options PREVIEW STAGES TEXT WIZARD

**Stage 1 (\$match)**

```
1 /**
2 * query: The query in MQL.
3 */
4 {
5   "directors": "John Landis",
6   "imdb.rating": {$gte:7.5,$lte:8.0}
7
8 }
```

Output after \$match stage (Sample of 4 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
```

```
_id: ObjectId('573a1397f29313caabce76f7')
plot: "Jake Blues, just out from prison, puts together his old band to save T..."
genres: Array (3)
runtime: 133
rated: "PG"
cast: Array (4)
num_mflix_comments: 1
```

```
_id: ObjectId('573a1397f29313caabce76f8')
fullplot: "Two Americans are on a walking tour and are...
imdb: Object
year: 1981
plot: "An American Werewolf in London"
genres: Array (2)
```

**Stage 2 (\$project)**

```
1 /**
2 * specifications: The fields to
3 * include or exclude.
4 */
5 {
6   "_id": 0,
7   "title": 1,
8   "imdb.rating": 1
9 }
```

Output after \$project stage (Sample of 4 documents)

```
imdb: Object
rating: 7.6
title: "Animal House"
```

```
imdb: Object
rating: 7.9
title: "The Blues Brothers"
```

```
imdb: Object
rating: 7.6
title: "An American Werewolf in London"
```

```
imdb: Object
rating: 7.5
title: "Trading Places"
```

## Exercise 2: Shows the location of all the theaters in the database on a map.

MongoDB Compass - mavi/sprint5.theaters

Compass

My Queries

CONNECTIONS (1) + ...

Search connect...

mavi > sprint5 > theaters

Documents 1.6K Aggregations Schema Indexes (1) Validation

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

EXPORT SCHEMA

document

geo

coordinates

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

theaterid

663 2860 1397 370 2931 547 1127 1455 844 581 8915 2955 580 146 1147 1953 412 8528

The screenshot shows the MongoDB Compass interface for a database named 'mavi' and a collection named 'theaters'. On the left, a sidebar lists connections and databases like 'mavi', 'admin', 'config', 'local', and 'sprint5'. Under 'sprint5', the 'theaters' collection is selected. The main area has tabs for 'Documents', 'Aggregations', 'Schema' (which is active), 'Indexes', and 'Validation'. A search bar at the top allows querying fields. Below it is an 'EXPORT SCHEMA' section with dropdowns for 'document' and 'geo'. A large map of North America displays numerous blue circular markers representing theater locations, primarily concentrated in the United States. At the bottom, a table shows the distribution of theater IDs across different ranges.