

TASK S5.01. Queries with MongoDB

LEVEL 1:

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

The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel lists the 'mavi' database with collections: admin, config, local, and 'sprint5'. The 'sprint5' database is selected, and its collections are listed: comments, movies, sessions, theaters, and users. The main panel displays the 'sprint5' database overview with the following data:

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
comments	4.10 kB	50 K	284.00 B	1	4.10 kB
movies	4.10 kB	24 K	1.60 kB	1	4.10 kB
sessions	4.10 kB	1	540.00 B	1	4.10 kB
theaters	4.10 kB	1.6 K	223.00 B	1	4.10 kB
users	4.10 kB	185	159.00 B	1	4.10 kB

A notification at the bottom left states: 'Import completed. 185 documents imported.'

Exercise 1:

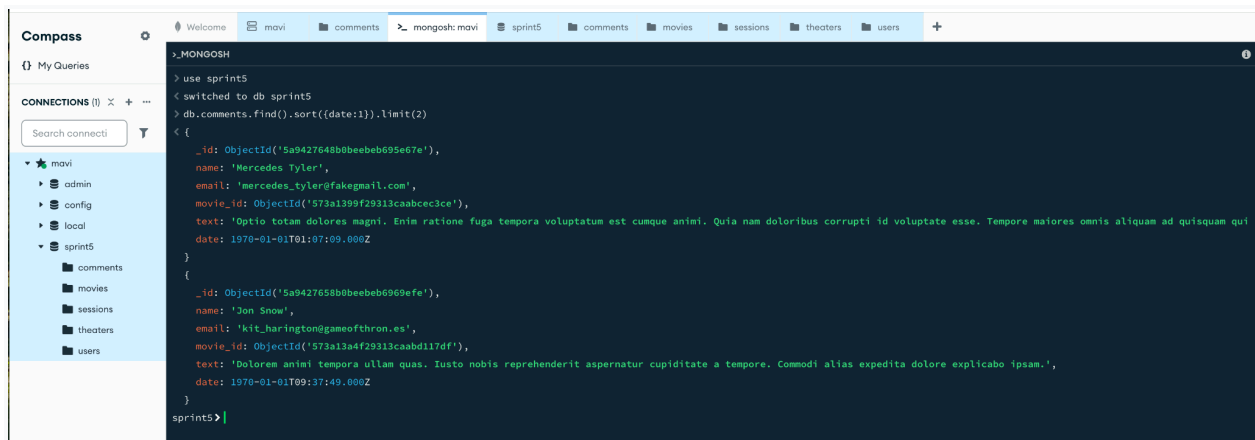
-Show the first 2 comments in the database.

The screenshot shows the MongoDB Compass interface with the 'sprint5' database selected. The 'comments' collection is selected, and the 'Documents' tab is active. The query bar shows the following query:

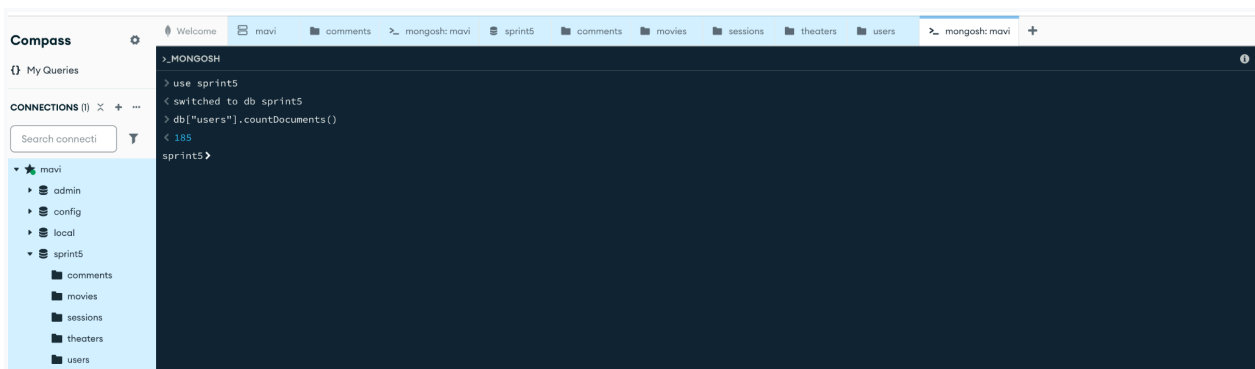
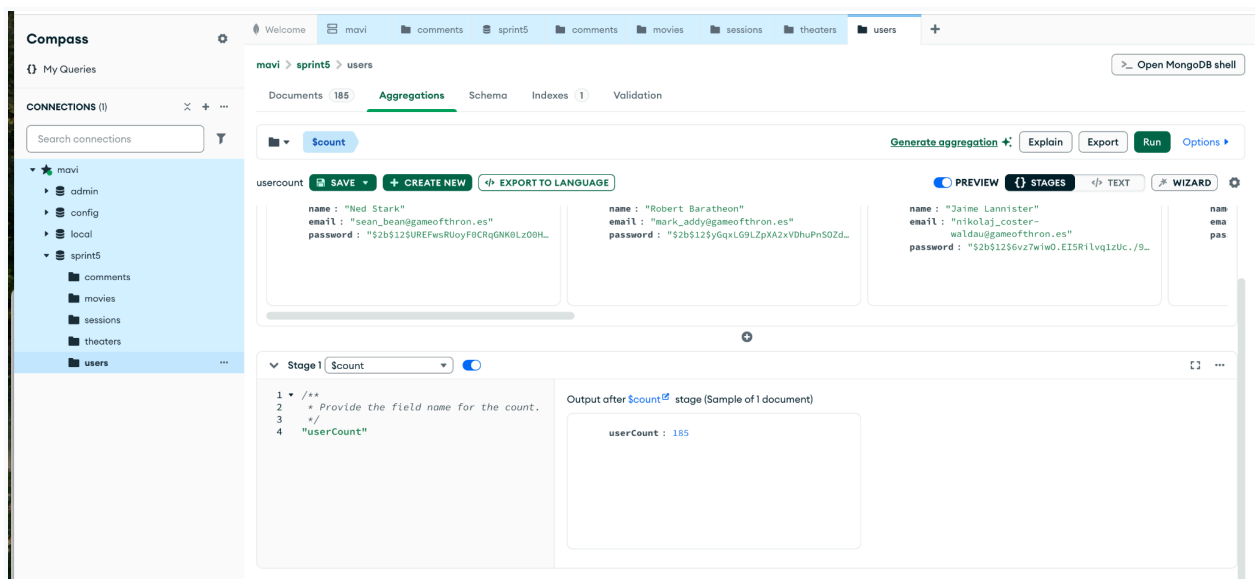
```
{}
```

The query results are displayed in a table with 2 documents:

Document
<pre>{ "_id": ObjectId("5a9427648b0beeb695e6fe"), "name": "Mercedes Tyler", "email": "mercedes_tyler@fakegmaill.com", "movie_id": ObjectId("573a1399f29313caabcec3ce"), "text": "Optio totam dolores magni. Enim ratione fuga temporam voluptatum est cu...", "date": 1978-01-01T01:07:09.000+00:00 }</pre>
<pre>{ "_id": ObjectId("5a9427654b0beeb695e6fe"), "name": "Jon Snow", "email": "kit_harlington@gameofthron.es", "movie_id": ObjectId("573a13a4f29313caabd117df"), "text": "Dolorem anini temporam ullam quas. Iusto nobis reprehenderit aspernatur...", "date": 1978-01-01T09:37:49.000+00:00 }</pre>



-How many users are registered?



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

The screenshot displays the MongoDB Compass interface. The top navigation bar shows the database 'mavi' and collection 'theaters'. The left sidebar lists the database structure, including collections like 'admin', 'config', 'local', 'sprint5', 'comments', 'movies', 'sessions', 'theaters', and 'users'. The main workspace is titled 'mavi > sprint5 > theaters' and shows an aggregation pipeline with two stages: '\$match' and '\$count'. The '\$match' stage is selected, showing a query in JSON: `{ "location.address.state": "CA" }`. The output for this stage shows a sample of 10 documents, with one document highlighted: `{ "_id": ObjectId('59a47286cf9a3a73e51e72e'), "theaterId": 1008, "location": { "address": { "state": "CA" } } }`. The '\$count' stage is also visible, showing a query: `{ "theaterCount": 1 }`. The output for this stage shows a sample of 1 document: `{ "theaterCount": 169 }`. The bottom panel shows the MongoDB shell with the following commands and output:

```
> 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

Connections: 185 Aggregations Schema Indexes 1 Validation

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("59b99db4cfa9a34dc7885b6"),
  "name": "Ned Stark",
  "email": "mar_kaddy@gameofthron.es",
  "password": "$2b$12$UREFwsRuoYf0CRqGNK8Lz09HML",
  "createDate": 2017-09-13T21:05:56.000+00:00 }
{ "_id": ObjectId("59b99db4cfa9a34dc7885b7"),
  "name": "Robert Baratheon",
  "email": "mar_kaddy@gameofthron.es",
  "password": "$2b$12$yGxLG9LzPxA2xVDhuPn50Zd...",
  "createDate": 2017-09-13T21:05:56.000+00:00 }
{ "_id": ObjectId("59b99db5cfa9a34dc7885b8"),
  "name": "Petyr Baelish",
  "email": "aidan_gillengameofthron.es",
  "password": "$2b$12$5qM.VvmeikyYYY7p7pK301cb...",
  "createDate": 2017-09-13T21:05:59.000+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("59b99db4cfa9a34dc7885b7"),
  "name": "Ned Stark",
  "email": "mar_kaddy@gameofthron.es",
  "password": "$2b$12$UREFwsRuoYf0CRqGNK8Lz09HML",
  "createDate": 2017-09-13T21:05:56.000+00:00 }
{ "_id": ObjectId("59b99db5cfa9a34dc7885b8"),
  "name": "Petyr Baelish",
  "email": "aidan_gillengameofthron.es",
  "password": "$2b$12$5qM.VvmeikyYYY7p7pK301cb...",
  "createDate": 2017-09-13T21:05:59.000+00:00 }
```

Stage 3: \$limit

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

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

```
{ "_id": ObjectId("59b99db4cfa9a34dc7885b6"),
  "name": "Ned Stark",
  "email": "mar_kaddy@gameofthron.es",
  "password": "$2b$12$UREFwsRuoYf0CRqGNK8Lz09HML",
  "createDate": 2017-09-13T21:05:56.000+00:00 }
```

-How many comedy movies are in our database?

Compass

Connections: 23.5K Aggregations Schema Indexes 1 Validation

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)

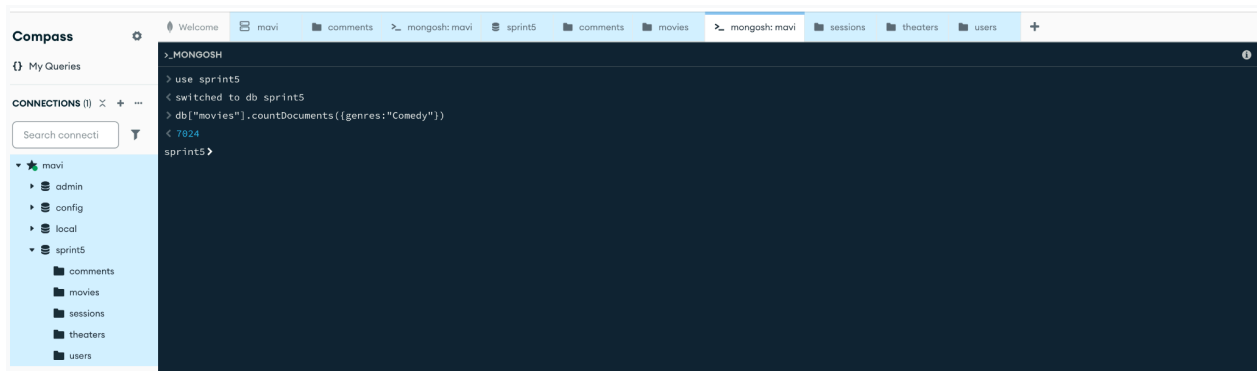
```
{ "_id": ObjectId("573a1390f29313caabdc4803"),
  "plot": "Cartoon figures announce, via comic strip balloons, that they will mov...",
  "genres": Array (3)
    0: "Animation"
    1: "Short"
    2: "Comedy"
  "runtime": 7
  "cast": Array (1) }
{ "_id": ObjectId("573a1390f29313caabdc50e5"),
  "plot": "The cartoonist, Winsor McCay, brings the Dinosaur back to life in th...",
  "genres": Array (3)
    0: "Animation"
    1: "Short"
    2: "Comedy"
  "runtime": 12
  "cast": Array (3) }
```

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 }
```



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

Stage 1: \$match

```

1 /**
2  * query: The query in MQL.
3  */
4 {
5   $match: {
6     "year": 1932,
7     $or: [
8       {"genres": "Drama"},
9       {"languages": "French"}
10    ]
11  }

```

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

```

{
  "_id": "ObjectId('573a1392f29313caabed99a3')",
  "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-...",
  "genres": Array (3),
  "runtime": 85,
  "cast": Array (4),
  "poster": "https://m.media-...",
  "title": "The Blue Light"
}

```

Stage 2: \$project

```

1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   $project: {
7     _id: 0,
8     title: 1,
9     year: 1,
10    genres: 1,
11    languages: 1
12  }

```

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

```

{
  "title": "The Blood of a Poet",
  "languages": Array (1),
  "year": 1932,
  "genres": Array (3),
  "languages": Array (2),
  "year": 1932
}

```

The screenshot shows the npass MongoDB interface. The left sidebar contains a file explorer with folders for 'admin', 'config', 'local', 'sprint5', 'comments', 'movies', 'sessions', 'theaters', and 'users'. The 'sprint5' folder is selected. The main window displays a MongoDB shell session with the following commands and output:

```
> 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'
  ],
  _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

Welcome | mavi | comments | sprint5 | comments | **movies** | sessions | theaters | users

My Queries

CONNECTIONS (1)

Search connections

mavi > sprint5 > movies

Documents 23.5K | **Aggregations** | Schema | Indexes (1) | Validation

Generate aggregation | Explain | Export | Run | Options

USAgawards | SAVE | CREATE NEW | EXPORT TO LANGUAGE | PREVIEW | STAGES | TEXT | WIZARD

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 }

```

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

```

{
  "_id": ObjectId('573a13ac729313caabd29366'),
  "fullplot": "The manager of the negative assets sector of Life magazine, Walter Mit...",
  "imdb": {
    "year": 2013,
    "plot": "When his job along with that of his co-worker are threatened, Walter t...",
    "genres": Array (3)
  }
}

```

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 \$project stage (Sample of 10 documents)

```

{
  "year": 2013,
  "title": "The Secret Life of Walter Mitty",
  "awards": {
    "wins": 6,
    "nominations": 13,
    "text": "6 wins & 13 nominations."
  },
  "countries": Array (2)
    0: "USA"
    1: "Canada"
  }
}

```

LEVEL 2:

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

npass

Welcome | mavi | **comments** | mongoosh: mavi | sprint5 | comments | movies | mongoosh: mavi | sessions | theaters | users

My Queries

CONNECTIONS (1)

Search connections

mavi > sprint5 > comments

Documents 50.3K | **Aggregations** | Schema | Indexes (1) | Validation

Generate aggregation | Explain | Export | Run | Options

Untitled - modified | SAVE | CREATE NEW | EXPORT TO LANGUAGE | PREVIEW | STAGES | TEXT | WIZARD

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": "toona_chapling@gameofthron.es",
  "movie_id": ObjectId('573a1390f29313caabd421...'),
  "text": "Rem itaque ad sit rem voluptatibus. Ad fugiat maxime illum optio iure ..."
}

```

Stage 2 | \$count

```

1 /**
2  * Provide the field name for the count.
3  */
4 {
5   "total_comments"
6 }

```

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

```

{
  "total_comments": 22841
}

```

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 MongoDB Compass interface with an aggregation pipeline applied to the 'theaters' collection. The pipeline consists of two stages: a \$match stage and a \$group stage.

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):

Document 1	Document 2	Document 3
<pre>{ "_id": ObjectId("59a47286cfa9a3a73e51e785"), "theaterId": 1092, "location": { "address": { "street1": "3100 14th St Nw", "city": "Washington", "state": "DC", "zipcode": "20010" } }, "geo": Object }</pre>	<pre>{ "_id": ObjectId("59a47287cfa9a3a73e51ec33"), "theaterId": 881, "location": { "address": { "street1": "4500 Wisconsin Ave Nw", "city": "Washington", "state": "DC", "zipcode": "20016" } }, "geo": Object }</pre>	<pre>{ "_id": ObjectId("59a47287cfa9a3a73e51ec33"), "theaterId": 8590, "location": { "address": { "street1": "50 Massachusetts Ave Nw", "city": "Washington", "state": "DC", "zipcode": "20002" } }, "geo": Object }</pre>

Stage 2: \$group

```
1 // **
2 * _id: The id of the group.
3 * fieldN: The first field name.
4 */
5 {
6   "_id": "$location.address.zipcode",
7   "theater_count": { $sum: 1 }
8 }
9 }
```

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

Document 1	Document 2	Document 3
<pre>{ "_id": "20010", "theater_count": 1 }</pre>	<pre>{ "_id": "20002", "theater_count": 1 }</pre>	<pre>{ "_id": "20010", "theater_count": 1 }</pre>

LEVEL 3:

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

Compass

Welcome | mavi | comments | sprint5 | comments | **movies** | sessions | theaters | users

My Queries

CONNECTIONS (1) | Search connect |

mavi > sprint5 > movies

Documents 23.5K | **Aggregations** | Schema | Indexes 1 | Validation

Smatch | \$project | Generate aggregation | Explain | Export | Run | Options

Untitled - modified | SAVE | CREATE NEW | EXPORT TO LANGUAGE | PREVIEW | STAGES | TEXT | WIZARD

Stage 1 | Smatch |

```
1 /**
2  * query: The query in MQL.
3  */
4
5 {
6   directors: "John Landis",
7   "imdb.rating" : { $gte: 7.5, $lte: 8.0 }
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. ..."`
- `_id: ObjectId('573a1397f29313caabce76f7')`
`plot: "Jake Blues, just out from prison, puts together his old band to save t..."`
- `_id: ObjectId('573a1397f29313caabce76f7')`
`fullplot: "Two Americans are on a walking tour and are..."`
- `_id: ObjectId('573a1397f29313caabce76f7')`
`fullplot: "Two Americans are on a walking tour and are..."`

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"`
- `title: "The Blues Brothers"`
`imdb: Object`
`rating: 7.9`
- `imdb: Object`
`rating: 7.6`
`title: "An American Werewolf in London"`
- `title: "Trading Places"`
`imdb: Object`
`rating: 7.5`

mpass

Welcome | mavi | comments | mongosh: mavi | sprint5 | comments | **movies** | mongosh: mavi | sessions | theaters | users

My Queries

CONNECTIONS (1) | Search connect |

mavi > sprint5 > movies

Documents 23.5K | **Aggregations** | Schema | Indexes 1 | Validation

Smatch | \$project | Edit | Explain | Export | Run | Options

Showing 1 - 4 count results

ALL RESULTS

- `imdb: Object`
`rating: 7.6`
`title: "Animal House"`
- `title: "The Blues Brothers"`
`imdb: Object`
`rating: 7.9`
- `imdb: Object`
`rating: 7.6`
`title: "An American Werewolf in London"`
- `title: "Trading Places"`
`imdb: Object`
`rating: 7.5`

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

