MongoDB Aggregation

Estimated time needed: **45** minutes

Objectives

After completing this lab, you will be able to:

- Describe simple aggregation operators that process and compute data such as $sort, $limit, $group, $sum, $min, $max, and $avg
- Combine operators to create multi-stage aggregation pipelines
- Build aggregation pipelines that draw insights about the data by returning aggregated values

About Skills Network Cloud IDE

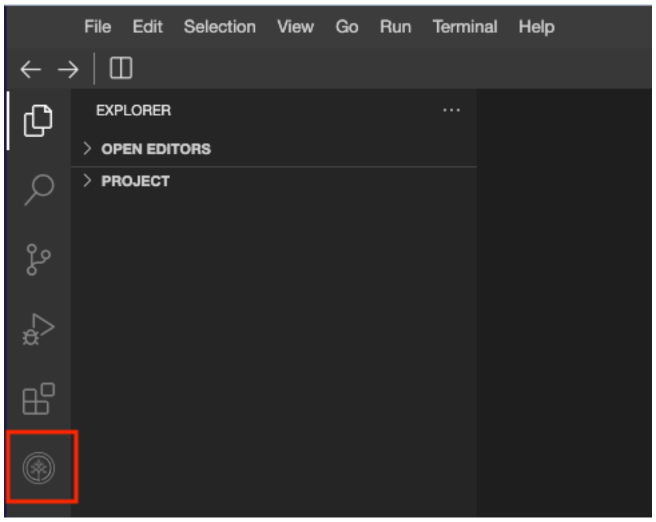
Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and MongoDB running in a Docker container.

Important Notice about this lab environment

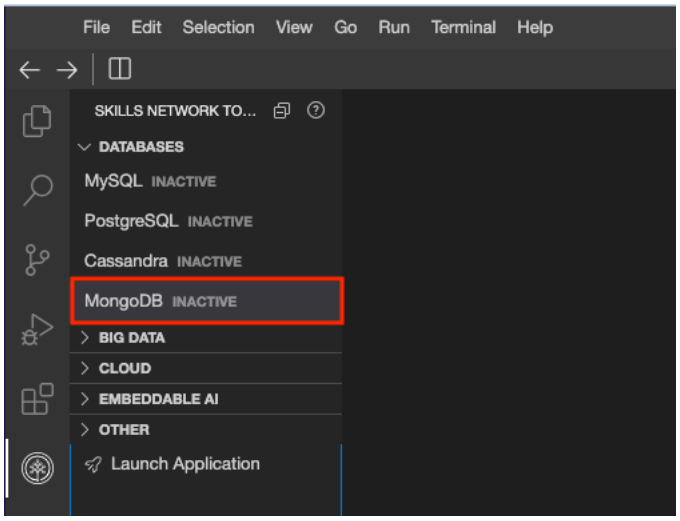
Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

Set-up: Start MongoDB

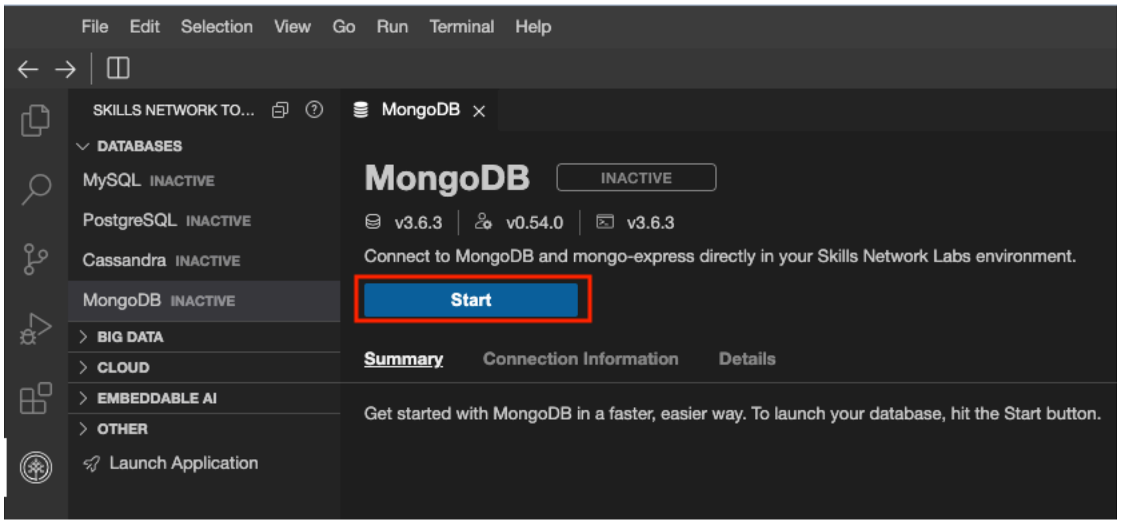
Navigate to Skills Network Toolbox.



You will notice MongoDB listed there, but inactive. Which means the database is not available to use.



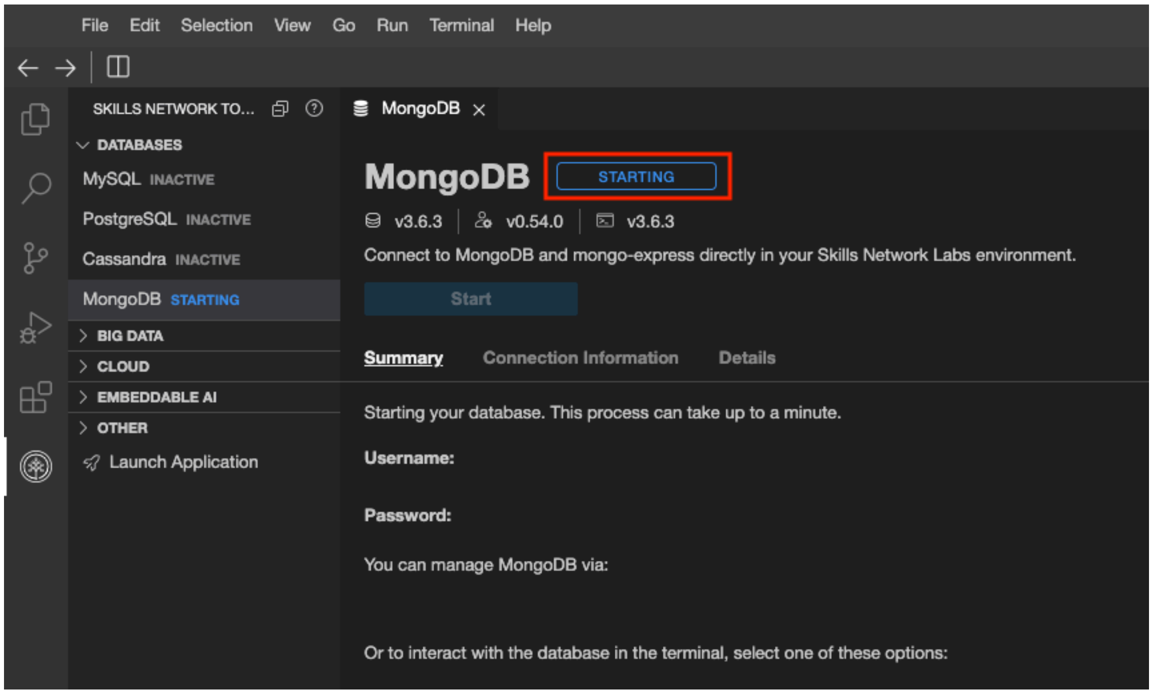
Once you click on it, you will see more details about it and a button to start it.



Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.

Get started with MongoDB in a faster, easier way. To launch your database, hit the Start button.

Clicking on the start button will run a background process to configure and start your MongoDB server.



Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.

Starting your database. This process can take up to a minute.

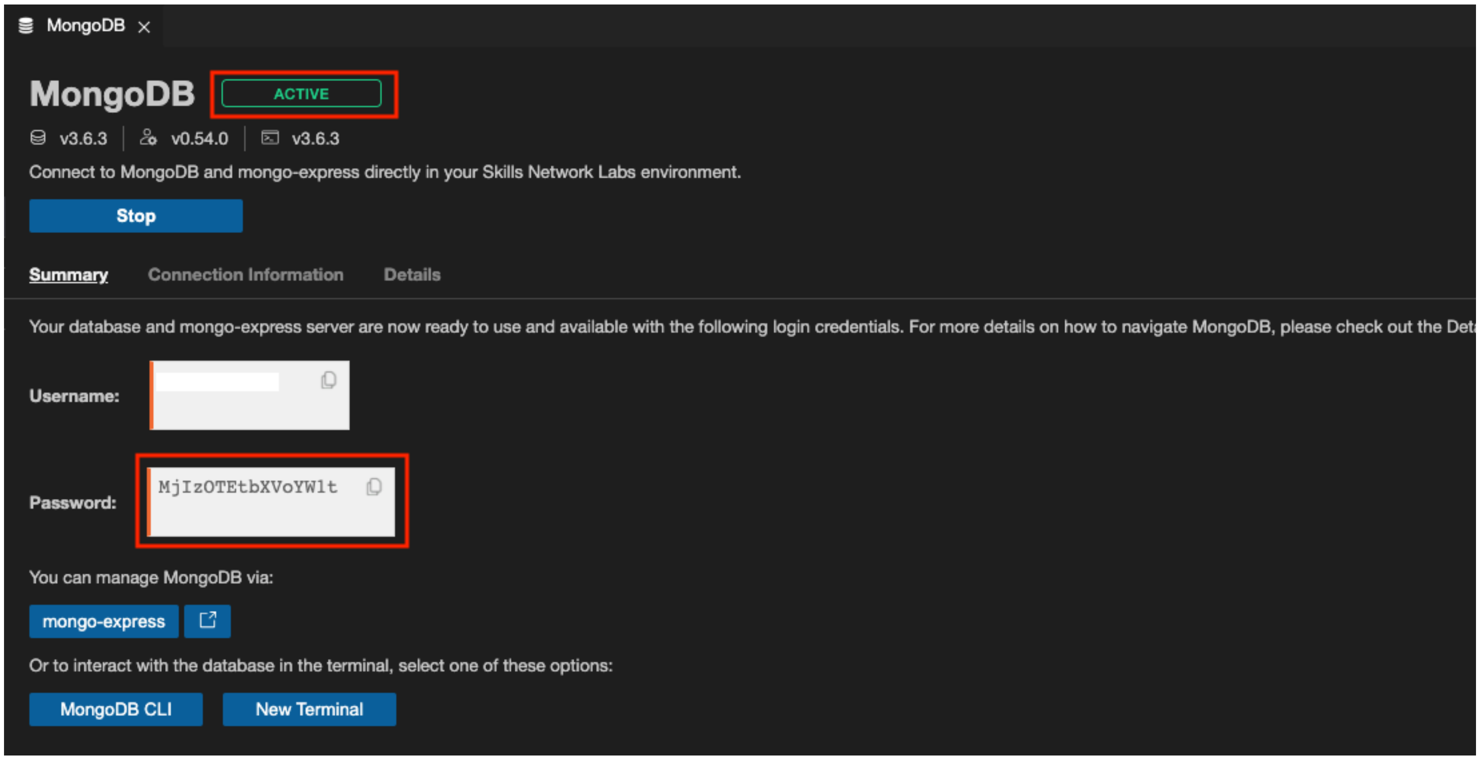
Username:

Password:

You can manage MongoDB via:

Or to interact with the database in the terminal, select one of these options:

Shortly after that, your server is ready for use. This deployment has access control enabled and MongoDB enforces authentication. So, take note of the password as you will need it to login as `root` user.


MongoDB ×

MongoDB ACTIVE

v3.6.3 | v0.54.0 | v3.6.3

Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.

Stop

Summary Connection Information Details

Your database and mongo-express server are now ready to use and available with the following login credentials. For more details on how to navigate MongoDB, please check out the Det

Username: [redacted]

Password: MjIzOTEtbXVoYW1t

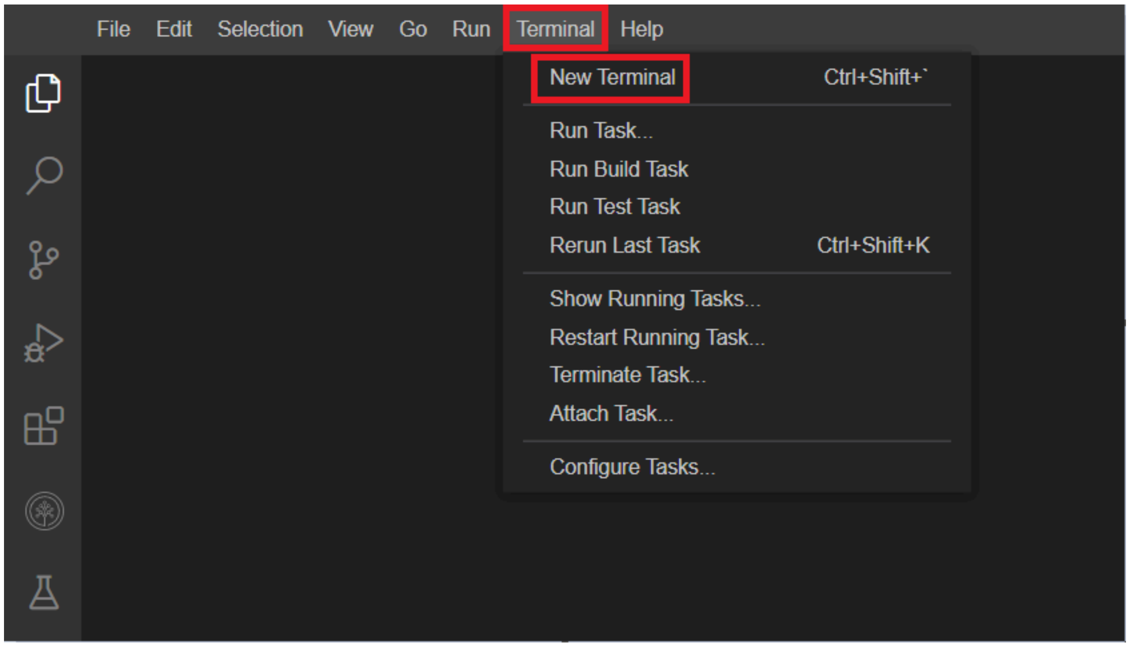
You can manage MongoDB via:

mongo-express

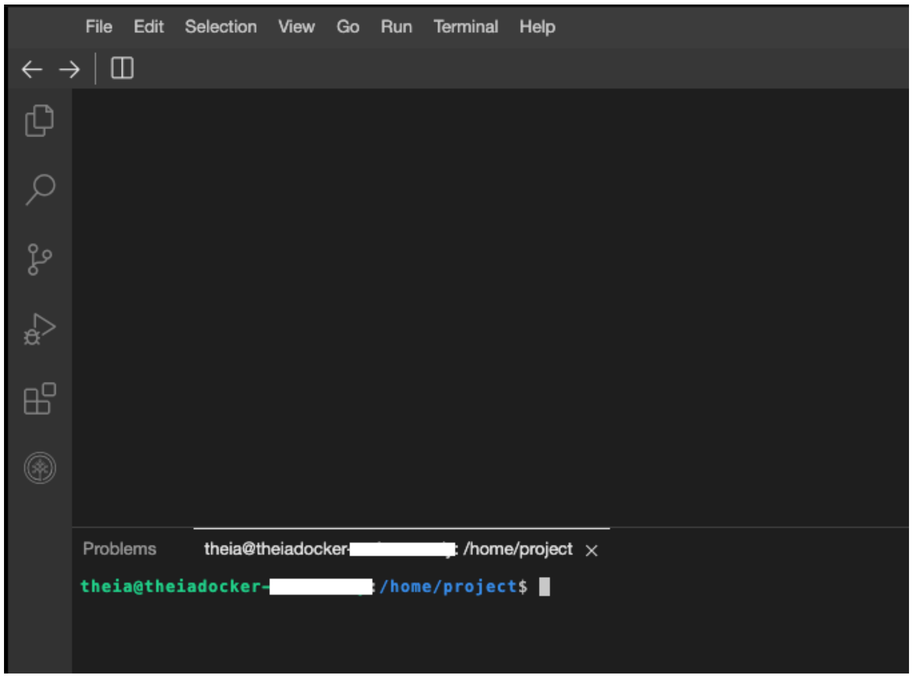
Or to interact with the database in the terminal, select one of these options:

MongoDB CLI New Terminal

You can now either open terminal and enter details yourself.



This will open a new terminal at the bottom of the screen as in the image below.



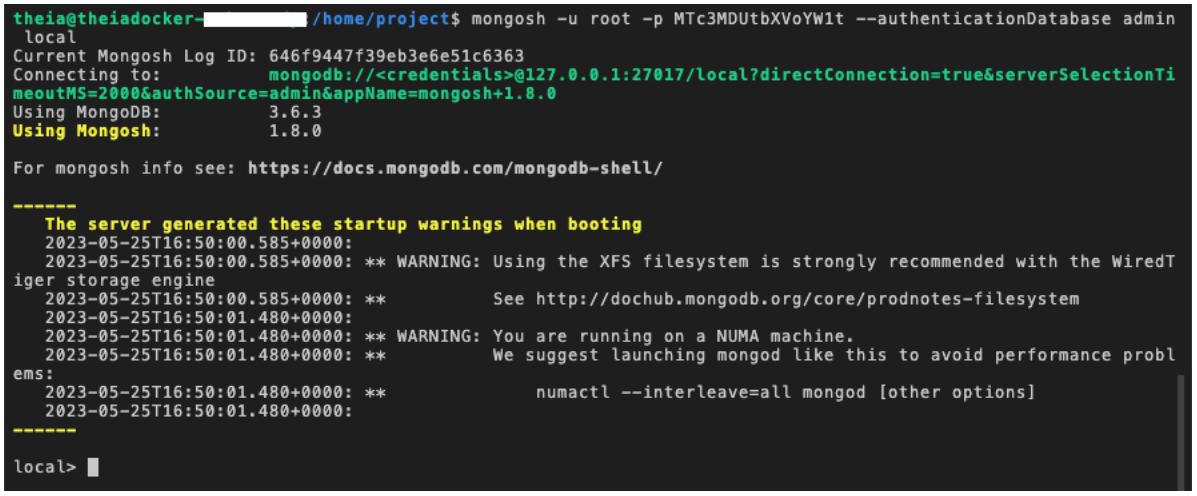
```
theia@theiadocker-[redacted]:/home/project$
```

Run the below command on the newly opened terminal. (You can copy the code by clicking on the little copy button on the bottom right of the codeblock below and then paste it, wherever you wish.)

```
1. 1
```

```
1. mongosh -u root -p PASSWORD --authenticationDatabase admin local
```

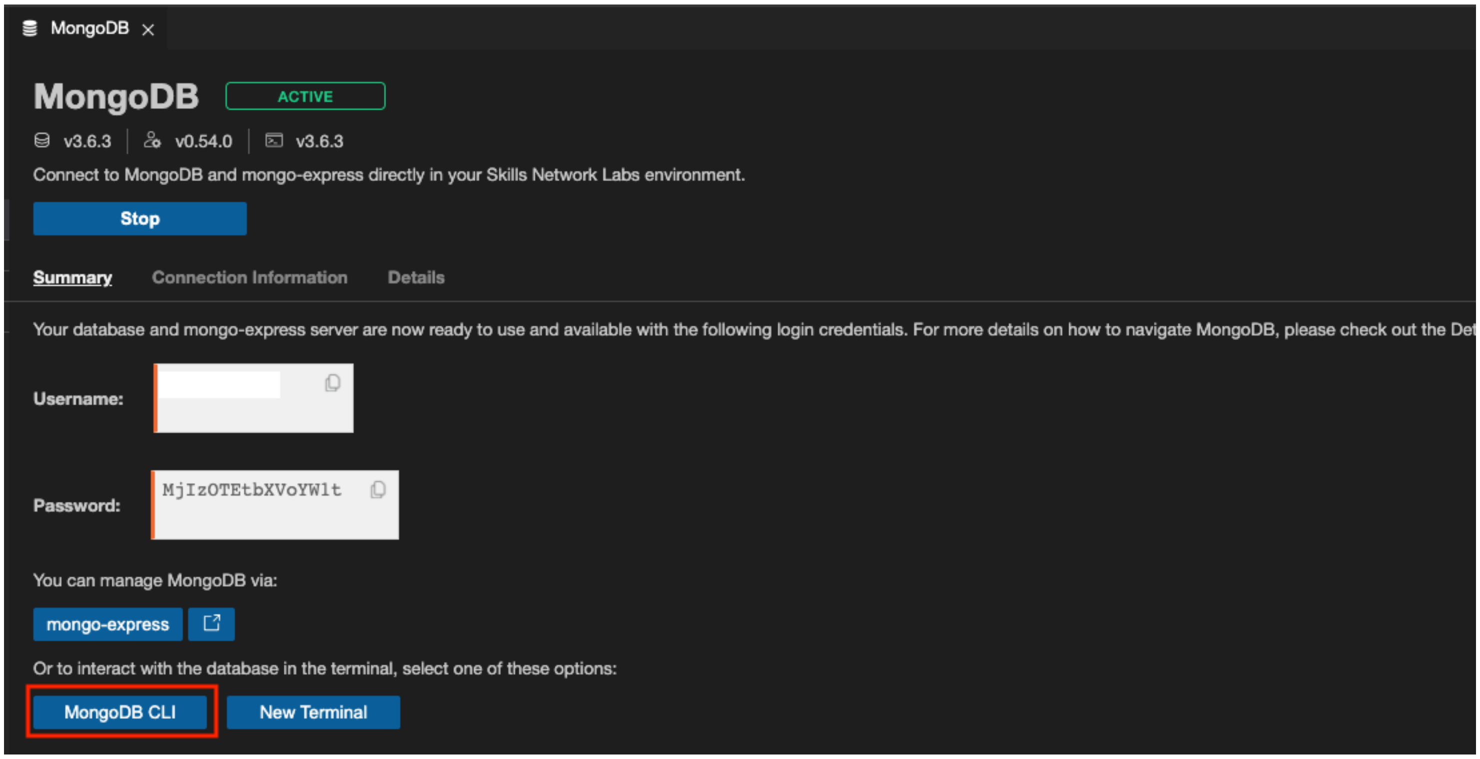
Copied! Executed!



```
theia@theiadocker-[redacted]/home/project$ mongosh -u root -p MTc3MDUtbXVoYW1t --authenticationDatabase admin
 local
Current Mongosh Log ID: 646f9447f39eb3e6e51c6363
Connecting to:          mongodb://<credentials>@127.0.0.1:27017/local?directConnection=true&serverSelectionTi
meoutMS=2000&authSource=admin&appName=mongosh+1.8.0
Using MongoDB:          3.6.3
Using Mongosh:          1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-05-25T16:50:00.585+0000:
   2023-05-25T16:50:00.585+0000: ** WARNING: Using the XFS filesystem is strongly recommended with the WiredT
iger storage engine
   2023-05-25T16:50:00.585+0000: **          See http://dochub.mongodb.org/core/prodnotes-filesystem
   2023-05-25T16:50:01.480+0000:
   2023-05-25T16:50:01.480+0000: ** WARNING: You are running on a NUMA machine.
   2023-05-25T16:50:01.480+0000: **          We suggest launching mongod like this to avoid performance probl
ems:
   2023-05-25T16:50:01.480+0000: **               numactl --interleave=all mongod [other options]
   2023-05-25T16:50:01.480+0000:
------

local>
```

The command contains the username and password to connect to mongodb server (the text after the -p option is the password). Your output would be different from the one shown above. Copy the command given to you, and keep it handy. You will need it in the next step.

Or you can simply click on MongoDB CLI which does that for you.


MongoDB ✕

MongoDB ACTIVE

v3.6.3 | v0.54.0 | v3.6.3

Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.

Stop

Summary   Connection Information   Details

Your database and mongo-express server are now ready to use and available with the following login credentials. For more details on how to navigate MongoDB, please check out the Det

Username: [redacted]

Password: MjIzOTEtbXVoYW1t

You can manage MongoDB via:

mongo-express

Or to interact with the database in the terminal, select one of these options:

MongoDB CLI   New Terminal

In MongoDB CLI (i.e. mongo shell), switch the context to `training` database.

```
use training
```

Copied!

And create a collection called `bigdata`

```
db.createCollection("bigdata")
```

Copied!

Exercise 1 - Load sample document

Load sample data into the `training` database in `marks` collection.

```
use training
db.marks.insert({"name":"Ramesh","subject":"maths","marks":87})
db.marks.insert({"name":"Ramesh","subject":"english","marks":59})
db.marks.insert({"name":"Ramesh","subject":"science","marks":77})
db.marks.insert({"name":"Rav","subject":"maths","marks":62})
db.marks.insert({"name":"Rav","subject":"english","marks":83})
db.marks.insert({"name":"Rav","subject":"science","marks":71})
db.marks.insert({"name":"Alison","subject":"maths","marks":84})
db.marks.insert({"name":"Alison","subject":"english","marks":82})
db.marks.insert({"name":"Alison","subject":"science","marks":86})
db.marks.insert({"name":"Steve","subject":"maths","marks":81})
db.marks.insert({"name":"Steve","subject":"english","marks":89})
db.marks.insert({"name":"Steve","subject":"science","marks":77})
db.marks.insert({"name":"Jan","subject":"english","marks":0,"reason":"absent"})
```

Copied!

Exercise 2 - Limiting the rows in the output

Using the `$limit` operator we can limit the number of documents printed in the output.

This command will print only 2 documents from the `marks` collection.

```
use training
db.marks.aggregate([{"$limit":2}])
```

Copied!

Exercise 3 - Sorting based on a column

We can use the `$sort` operator to sort the output.

This command sorts the documents based on field marks in ascending order.

```
db.marks.aggregate([{"$sort":{"marks":1}}])
```

Copied!

This command sort the documents based on field marks in descending order.

```
db.marks.aggregate([{"$sort":{"marks":-1}}])
```

Copied!

Exercise 4 - Sorting and limiting

Aggregation usually involves using more than one operator.
A pipeline consists of one or more operators declared inside an array.
The operators are comma separated.
Mongodb executes the first operator in the pipeline and sends its output to the next operator.

Let's create a two stage pipeline that answers the question "What are the top 2 marks?”

```
db.marks.aggregate([
    {"$sort":{"marks":-1}},
    {"$limit":2}
])
```

Copied!

Exercise 5 - Group by

The operator `$group` by, along with operators like $sum, $avg, $min, $max, allows us to perform grouping operations.

This aggregation pipeline prints the average marks across all subjects.

```
db.marks.aggregate([
{
    "$group":{
        "_id":"$subject",
        "average":{"$avg":"$marks"}
    }
}
])
```

Copied!

The above query is equivalent to the below sql query.

```
SELECT subject, average(marks)
FROM marks
```

```
GROUP BY subject
```

Copied!

Exercise 6 - Putting it all together

Now let's put together all the operators we have learnt to answer the question. "Who are the top 2 students by average marks?”

This involves:

- finding the average marks per student
- sorting the output based on average marks in descending order
- limiting the output to two documents

```

db.marks.aggregate([
    {
        "$group":{
            "_id":"$name",
            "average":{"$avg":"$marks"}
            }
    },
    {
        "$sort":{"average":-1}
    },
    {
        "$limit":2
    }
])
```

Copied!

Practice exercises

1. Problem: Find the total marks for each student across all subjects.

▶ Click here for Hint
▶ Click here for Solution

2. Problem: Find the maximum marks scored in each subject.

▶ Click here for Hint
▶ Click here for Solution

3. Problem: Find the minimum marks scored by each student.

▶ Click here for Hint
▶ Click here for Solution

4. Problem: Find the top two subjects based on average marks.

▶ Click here for Hint
▶ Click here for Solution

Summary

In this lab, you have gained an understanding of Aggregation pipelines in MongoDB.

Author(s)

Muhammad Yahya