

Final Assignment: Data Engineering for a Consulting Firm



Estimated time needed: 45 mins

You are a data engineer at a data analytics consulting company. Your company prides itself in being able to efficiently handle data in any format on any database on any platform. Analysts in your office need to work with data on different databases, and data in different formats. While these analysts are good at analyzing data, they count on you to be able to move data from external sources into various databases, to be able to move data from one type of database to another, and be able to run basic queries on various databases.

Objectives

In this assignment you will:

- Import data into a MongoDB database.
- Query data in a MongoDB database.
- Export data from MongoDB.
- Import data into a Cassandra database.
- Query data in a Cassandra database.

About Skills Network Cloud IDE

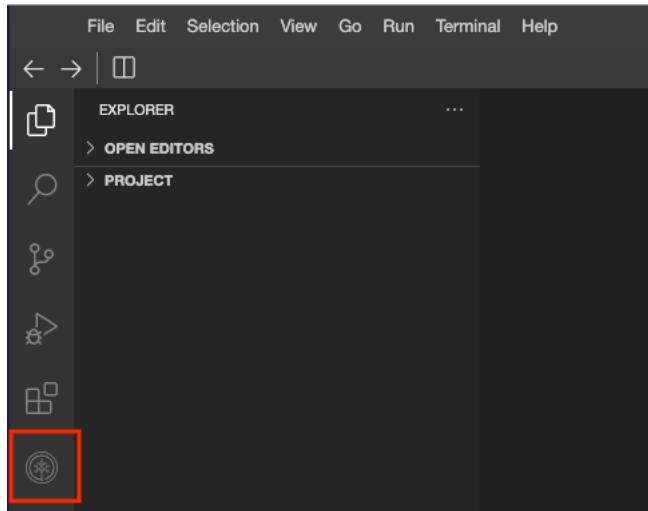
Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based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

Important Notice about this lab environment

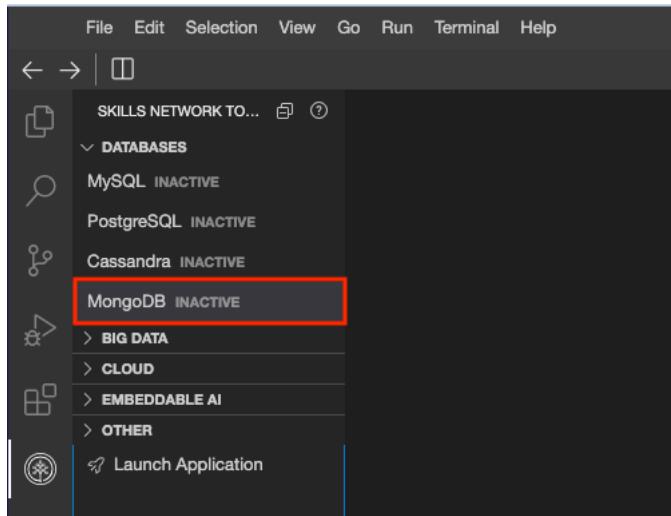
Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete these labs in a single session, to avoid losing your data.

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 the database, you will see more details and a button to start the database.

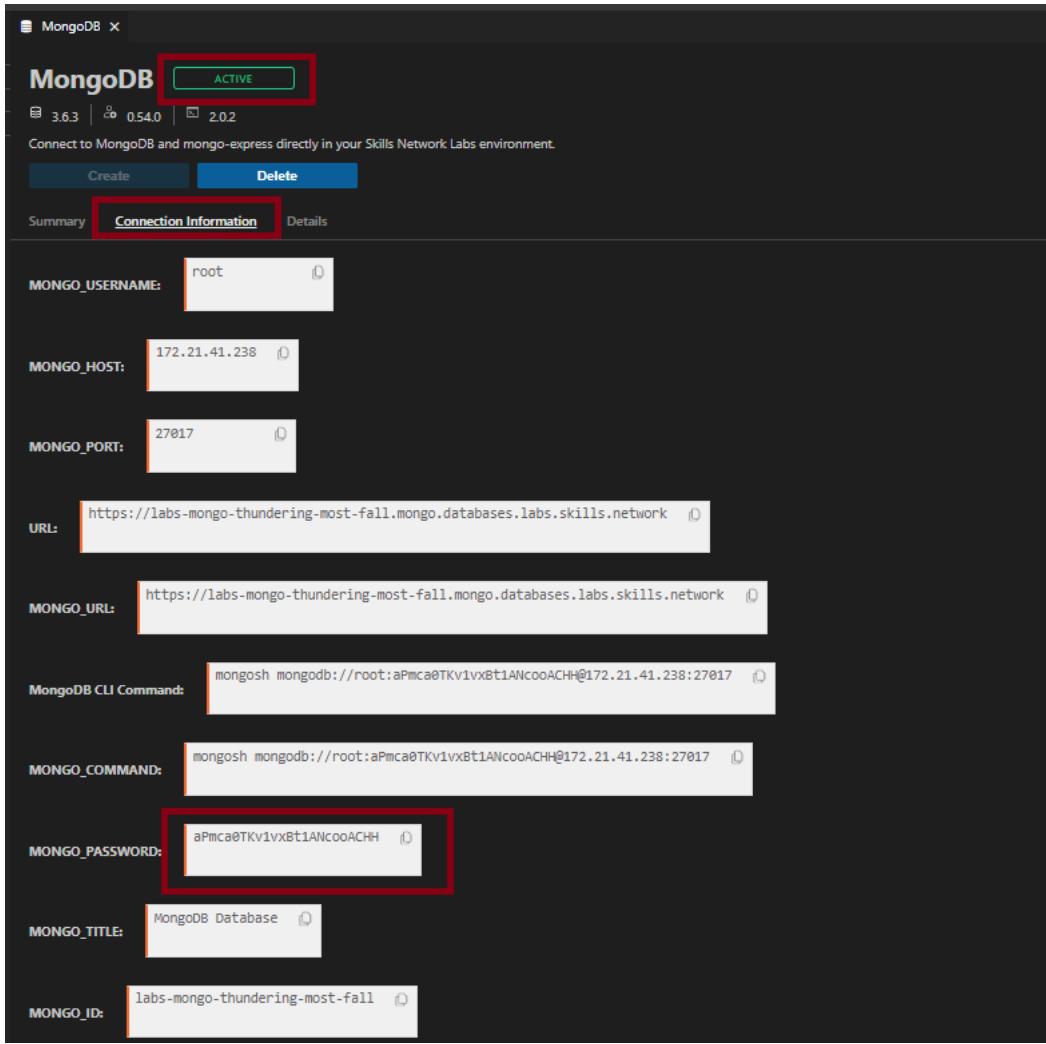
The screenshot shows the MongoDB database details page. The 'MongoDB' tab is selected. The status is 'INACTIVE'. Below it, the version is 3.6.3, mongo version is 0.54.0, and mongo-express version is 2.0.2. A message says 'Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.' Below the message are 'Create' and 'Delete' buttons, with 'Create' highlighted by a red box. At the bottom, there are tabs for 'Summary', 'Connection Information', and 'Details', and a note: 'Get started with MongoDB in a faster, easier way. To launch your database, hit the Start button.'

Clicking the Create button runs a background process to configure and start your MongoDB server.

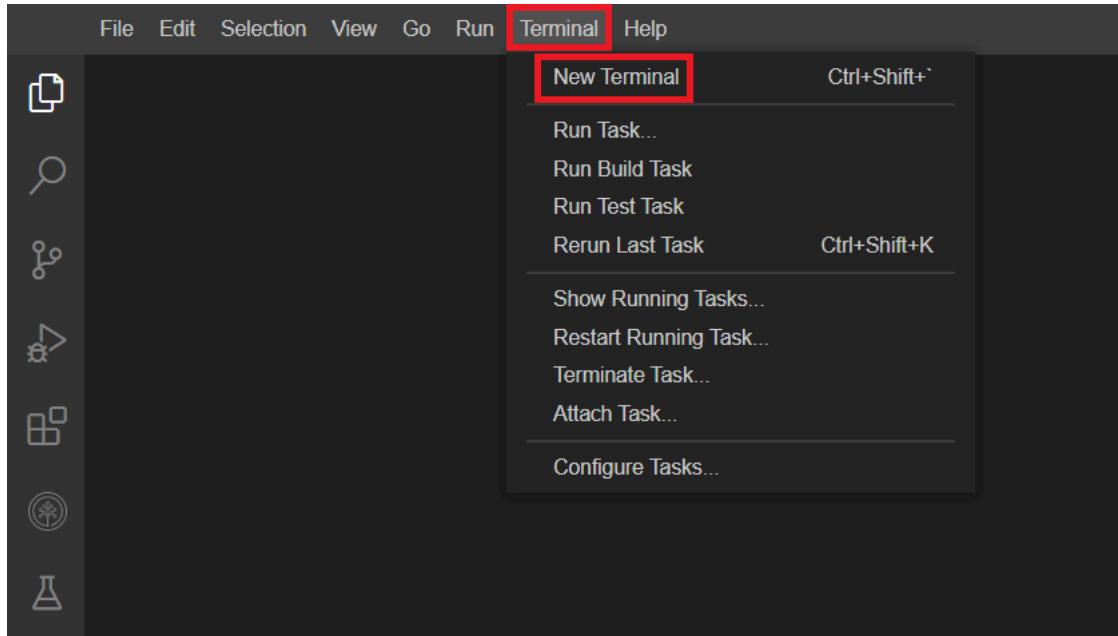
The screenshot shows the MongoDB database details page again, but the status has changed to 'STARTING'. The 'MongoDB' tab is still selected. The version information remains the same. A message says 'Starting your database. This process can take up to a minute.' Below the message are fields for 'Username:' and 'Password:', and a note: 'You can manage MongoDB via:'. At the bottom, there is a note: 'Or to interact with the database in the terminal, select one of these options:'.

Soon, your server is ready for use. This deployment has access control enabled and MongoDB enforces authentication. So, take note of the password. You will need this password to login as root user.

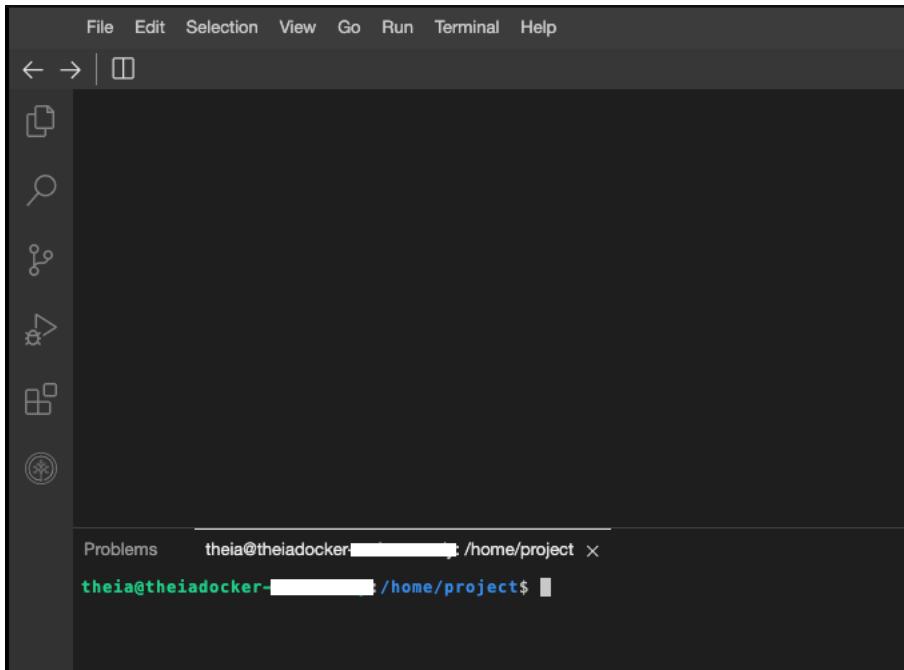
Note: For Password and other information click on Connection Information



You can now open a new terminal window.



The new terminal window opens at the bottom of the screen as seen in the following image.



Run the following command in the newly opened terminal window. (You can copy the code by clicking on the copy button on the bottom right of the following codeblock and then paste the code where needed.)

```
mongosh -u root -p PASSWORD --authenticationDatabase admin local --host mongo
```

```
theia@theiadocker-nikeshkr:/home/project$ mongosh -u root -p aPmca0TKv1vxBt1ANcooAChH --authenticationDatabase admin local --host mongo
Current Mongosh Log ID: 66de8a0693cc32c77c5e739b
Connecting to:      mongodb://<credentials>@mongo:27017/local?directConnection=true&authSource=admin&appName=mongosh+2.3.0
Using MongoDB:      3.6.3
Using Mongosh:      2.3.0

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

-----
The server generated these startup warnings when booting
2024-09-09T04:08:06.675+0000:
2024-09-09T04:08:06.675+0000: ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2024-09-09T04:08:06.675+0000: ** See http://dochub.mongodb.org/core/prodnotes-filesystem
-----
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 output shown in the prior image. Copy the command given to you, and keep this command handy. You will need this command in the next step.

Or, you can simply click on MongoDB CLI

The screenshot shows a MongoDB configuration interface. At the top, there's a header with the MongoDB logo and a status bar indicating it's 'ACTIVE'. Below this, there are three tabs: 'Create', 'Delete', and 'Summary' (which is currently selected). A note below the tabs says, 'Connect to MongoDB and mongo-express directly in your Skills Network Labs environment.' There are two large blue buttons at the bottom: 'mongo-express' and 'New Terminal'. The 'MongoDB CLI' button is highlighted with a red box.

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

```
use training
```

Create a collection called `bigdata`

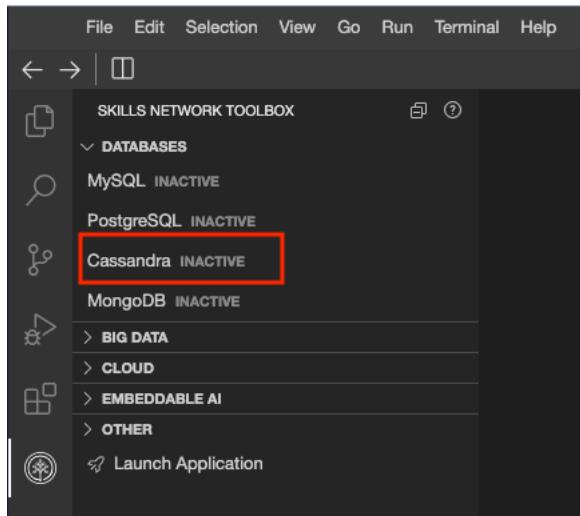
```
db.createCollection("bigdata")
```

Set-up: Start Cassandra

Navigate to Skills Network Toolbox.

The screenshot shows the Skills Network Toolbox interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. On the left, there's a vertical sidebar with icons for Explorer, Open Editors, Project, and a search function. The Cassandra icon, which is a globe with a grid, is highlighted with a red box.

Cassandra is listed, but inactive, which means a database is not available to use.



Once you click on the database, you will see more details and the button to start the database.

This screenshot shows the detailed view for the Cassandra database. The left sidebar remains the same. The main area displays the 'Cassandra' entry with the status 'INACTIVE'. It shows two versions: v4.1 and v5.0.1. A large blue 'Create' button is prominently displayed, which is also highlighted with a red box. Below the button, there are tabs for 'Summary', 'Connection Information', and 'Details'. A note below the tabs says, 'Get started with Cassandra in a faster, easier way. To launch your database, hit the Start button.' At the bottom, there is a section for entering a 'Username:' and 'Password:', followed by a note about interacting with the database via the terminal.

Clicking the Create button runs a background process to configure and start your Cassandra server.

This screenshot shows the same interface after the 'Create' button was clicked. The 'Cassandra' entry now has the status 'STARTING', indicated by a red box around the status text. The rest of the interface remains the same, including the 'Create' button, version information, and connection details.

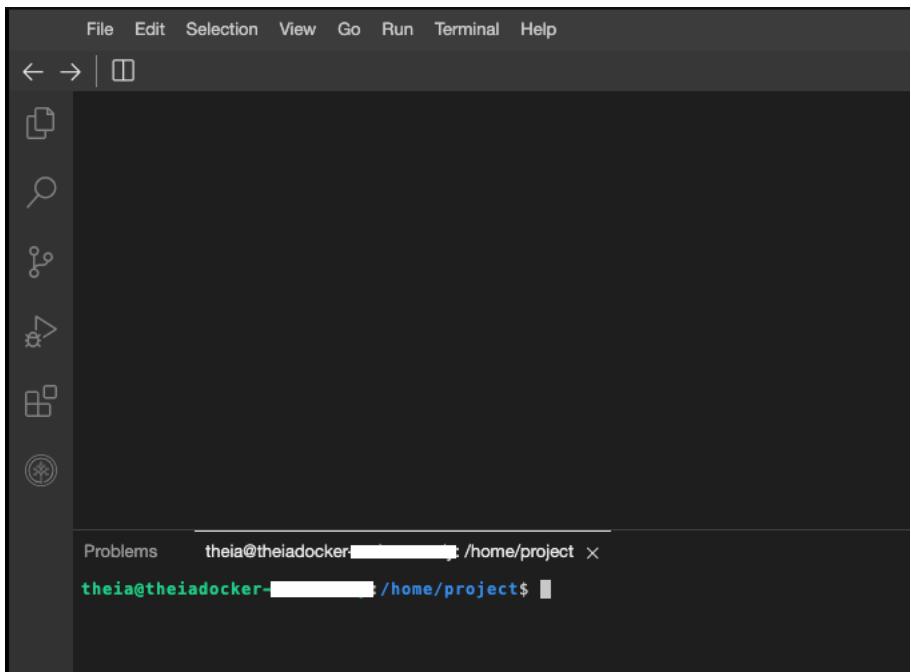
Shortly after that, your server is ready for use. This deployment has access control enabled and Cassandra enforces authentication. Click on the Connection Information take note of the Cassandra CLI Command as you will need to login as a root user.

The screenshot shows the Skills Network Labs interface for managing databases. At the top, there are tabs for MongoDB and Cassandra, with Cassandra being the active tab. Below the tabs, it says "Cassandra" and "ACTIVE". There are two versions listed: v4.1 and v5.0.1. A message encourages connecting directly to Cassandra. Below this, there are "Create" and "Delete" buttons. The main area has three tabs: "Summary", "Connection Information" (which is selected and highlighted with a red box), and "Details". Under "Connection Information", there are fields for "Username" (nikeshkr), "Password" (MTkyMS1uaWt1c2hr), "Host" (127.0.0.1), and "Port" (9042). At the bottom, there is a "Cassandra CLI Command" field containing the command: `cqlsh 172.21.151.104 9042 --username root --password 8VoOKNCYFggUh59uvLNz1VD`.

You can now type \open terminal and enter the details.

The screenshot shows the Skills Network Labs interface with a focus on the "Terminal" menu. The "Terminal" menu item is highlighted with a red box. A dropdown menu is open, showing options: "New Terminal" (highlighted with a red box), "Run Task...", "Run Build Task", "Run Test Task", "Rerun Last Task" (with a keyboard shortcut "Ctrl+Shift+K"), "Show Running Tasks...", "Restart Running Task...", "Terminate Task...", "Attach Task...", and "Configure Tasks...". On the left side of the interface, there is a vertical toolbar with various icons.

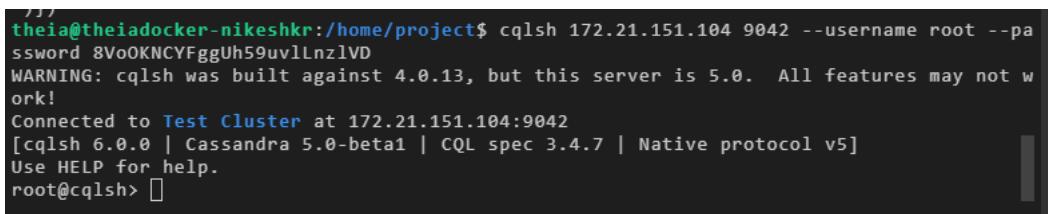
This action opens a new terminal at the bottom of the screen as seen in the following image.



The screenshot shows a terminal window within a code editor interface. The terminal bar at the bottom has tabs for 'Problems' and 'theia@theiadocker-[REDACTED]:~/home/project' with an 'X' icon. The main terminal area displays a single line of text: 'theia@theiadocker-[REDACTED]:~/home/project\$'. The background of the terminal area is dark gray, and the text is white.

Run the following command in 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 the code where needed.)

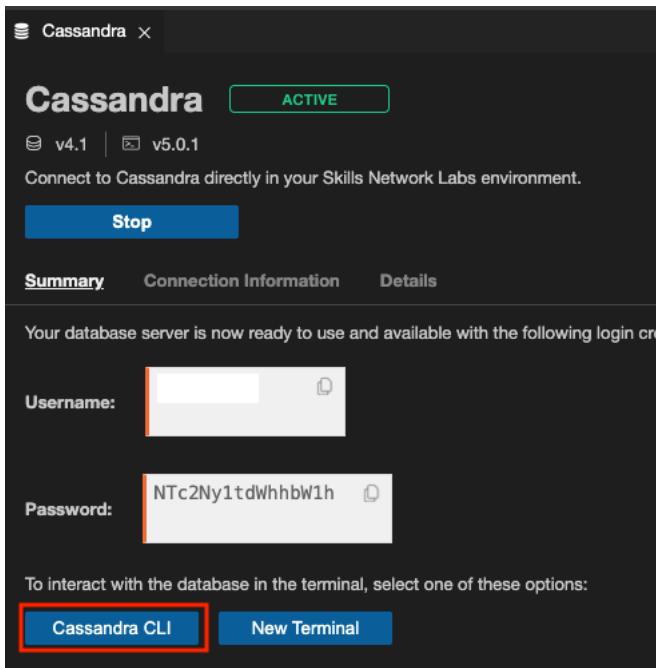
```
cqlsh HOST PORT --username root --password PASSWORD
```



```
theia@theiadocker-nikeshkr:/home/project$ cqlsh 172.21.151.104 9042 --username root --password 8VoOKNCYFggUh59uvLlnz1VD
WARNING: cqlsh was built against 4.0.13, but this server is 5.0. All features may not work!
Connected to Test Cluster at 172.21.151.104:9042
[cqlsh 6.0.0 | Cassandra 5.0-beta1 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
root@cqlsh> 
```

This command contains the username and password to connect to Cassandra server. Your output could be different from the one shown above. Copy the command given to you, and keep the command handy. You will use this command in the next step.

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



Exercise 1: Working with a MongoDB database

Download sample data file

Use the following command to download the data file in your Cloud IDE project directory.

```
curl -O https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-DB0151EN-edX/labs/FinalProject/movies.json
```

A sample movie document

```
{
  _id: '9',
  title: 'The Lost City of Z',
  genre: 'Action,Adventure,Biography',
  Description: 'A true-life drama, centering on British explorer Col. Percival Fawcett, who disappeared while searching for a mysterious city in the Amazon rainforest in 1925. The film follows his son, who sets out to find his father and uncovers a dark secret about the lost city of Z that has been hidden for decades.',
  Director: 'James Gray',
  Actors: 'Charlie Hunnam, Robert Pattinson, Sienna Miller, Tom Holland',
  year: 2016,
  'Runtime (Minutes)': 141,
  rating: 'unrated',
  Votes: 7188,
  'Revenue (Millions)': 8.01,
  Metascore: 78
}
```

Task 1: Import movies.json into mongodb server into a database named entertainment and a collection named movies.

Now import the data in movies.json and take a screenshot of the command you used and the output.

```
theia@theiadocker-nikeshkr:/home/project$ mongoimport -u root -p VxUlDDxYdFvFGiEVhKMqrgRg --authenticationDatabase admin -d entertainment -c movies --host mongo movies.json
2024-08-30T11:14:09.987-0400      connected to: mongodb://mongo/
2024-08-30T11:14:10.012-0400    100 document(s) imported successfully. 0 document(s) failed to import.
```

Assessment: Take a screen capture of the output and save the screen capture as 01-mongo-import.png (Save the images using either a .jpg or .png extension).

Task 2: Write a mongodb query to find the year in which most number of movies were released

▼ Click here for Hint

The \$group stage can be used to group documents by a certain field. You can calculate the count of movies within the group using the \$sum aggregation operator. Your query should:

- Group movies by their release year.
- Calculate the total count of movies for each year
- Sort the years in descending order of movie count
- Limit the output to 1 document (year) which has the highest movie count.

```
db.movies.aggregate([
  {
    "$group": {
      "_id": "$year",
      "movieCount": { "$sum": 1 }
    }
  }
])
```

Take a screenshot of the command you used and the output showing 73 movies in year 2016.

```
entertainment> db.movies.aggregate([
...   {
...     "$group": {
...       "_id": "$year",
...       "movieCount": { "$sum": 1 }
...     }
...   },
...   {
...     "$sort": { "movieCount": -1 }
...   },
...   {
...     "$limit": 1
...   }
... ])
[ { _id: 2016, movieCount: 73 } ]
entertainment>
```

Assessment: Take a screen capture of the output and save the screen capture as 02-most-movies-year.png (You can save the image using either the .jpg or .png extension).

Task 3: Write a mongodb query to find the count of movies released after the year 1999

Take a screenshot of the command you used and the output showing 99 movies after year 1999.

```
entertainment> db.movies.countDocuments({ year: { $gt: 1999 } })
99
```

Assessment: Take a screen capture of the output and save the screen capture as 03-movies-count-1999.png (You can save the image using either the .jpg or .png extension).

Task 4: Write a query to find out the average votes for movies released in 2007

▼ Click here for Hint

use the \$match operator to filter for movies released in 2007. And \$group with \$avg operator to find average votes.

```
db.books.aggregate([
  { $match : "filter criteria" },
  { $avg: { _id: "$field", averageVotes: "syntax for $avg" } }
])
```

Take a screenshot of the command you used and the output.

```
entertainment> db.movies.aggregate([
...   {
...     $match: { year: 2007 }
...   },
...   {
...     $group:
...     {
...       _id: "$year",
...       averageVotes: { $avg: "$Votes" }
...     }
...   }
... ],
... { _id: 2007, averageVotes: 192.5 } ])
```

Assessment: Take a screen capture of the output and save the screen capture as 04-average-votes.png You can save the image using either the.jpg or .png extension).

Task 5: Export the fields `_id`, `title`, `year`, `rating` and `director` from the `movies` collection into a file named `partial_data.csv`

Take a screenshot of the command you used and its output.

The terminal window shows the following steps:

- A CSV file named `partial_data.csv` is displayed, containing 30 rows of movie data with columns: `_id`, `title`, `year`, `rating`, and `director`.
- The command `mongosh mongodb://<credentials>@172.21.157.39:27017/?directConnection=true` is run to connect to the MongoDB database.
- The command `theia@theiadocker-nikeshkr:/home/project$ mongoexport -u root -p abDj5HeED04oRvvJMQs4mjqo --authenticationDatabase admin -d entertainment -c movies -f "_id,title,year,rating,director" --type=csv -o partial_data.csv --host mongo` is run to export the data.
- The output shows the connection to the MongoDB database and the successful export of 100 records.

Assessment: Take a screen capture of the output and save the screen capture as 05-mongo-export.png (images can be saved with either .jpg or .png extension).

Exercise 2 - Working with a Cassandra database

Download sample data file

If you haven't managed to successfully export data into `partial_data.csv` file, then download the data file ready created for you with the following command.

```
curl -O https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-DB0151EN-edX/labs/FinalProject/partial_data.csv
```

Task 6 - Create a keyspace named `entertainment`

▼ Click here for Hint

use the CREATE KEYSPACE command

```
CREATE KEYSPACE keyspace_name
WITH replication replication_details
```

Once created, use the describe command and use the output for assessment.

```
cassandra@cqlsh> CREATE KEYSPACE entertainment
... WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};

Warnings :
Your replication factor 3 for keyspace entertainment is higher than the number of nodes 1

cassandra@cqlsh> describe keyspaces
entertainment    system_auth      system_schema   system_views
system          system_distributed system_traces  system_virtual_schema
```

Assessment: Take a screen capture of the output and save the screen capture as 06-describe-keyspaces.png (You can save the image using either the .jpg or .png extension).

Task 7 - Import partial_data.csv into cassandra server into a keyspace named entertainment and a table named movies

While creating the table movies configure all of the columns as text columns including the id column.

- _id
- title
- year
- rating
- director

▼ Click here for Hint

use the CREATE TABLE command

```
CREATE TABLE table_name(
    field_name text PRIMARY KEY,
    second_field_name text
);
```

```
cassandra@cqlsh:entertainment> CREATE TABLE movies(
    ...     id text PRIMARY KEY,
    ...     title text,
    ...     year text,
    ...     rating text,
    ...     director text
    ... );
cassandra@cqlsh:entertainment> describe movies
CREATE TABLE entertainment.movies (
    id text PRIMARY KEY,
    director text,
    rating text,
    title text,
    year text
) WITH additional_write_policy = '99p'
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND cdc = false
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '16'}
AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND memtable = 'default'
AND crc_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair = 'BLOCKING'
AND speculative_retry = '99p';
```

And now import the data present in `partial_data.csv`

▼ Click here for Hint

use the `COPY` command

```
COPY keyspace.table_name(columns) FROM 'file_path' WITH DELIMITER=',' AND HEADER=TRUE;
```

```
cassandra@cqlsh:entertainment> COPY entertainment.movies(id,title,year,rating,director) FROM '/home/project/partial_data.csv' WITH DELIMITER=',' AND HEADER=TRUE;
Using 15 child processes
Starting copy of entertainment.movies with columns [id, title, year, rating, director].
Processed: 100 rows; Rate: 25 rows/s; Avg. rate: 47 rows/s
100 rows imported from 1 files in 0 day, 0 hour, 0 minute, and 2.116 seconds (0 skipped).
```

Assessment: Take a screen capture of the output and save the screen capture as `07-movies-imported.png` (You can save the image using either the `.jpg` or `.png` extension).

Task 8 - Write a cql query to count the number of rows in the `movies` table

▼ Click here for Hint

use the `SELECT COUNT(*)` command

```
SELECT COUNT(*) FROM table_name;
```

```
cassandra@cqlsh:entertainment> SELECT COUNT(*) FROM movies;
count
-----
100
(1 rows)
Warnings:
Aggregation query used without partition key
```

Assessment: Take a screen capture of the output and save the screen capture as `08-movies-count.png` You can save the image using either the `.jpg` or `.png` extension).

Task 9 - Create an index for the rating column in the `movies` table using cql

▼ Click here for Hint

use the `CREATE INDEX` command

```
CREATE INDEX IF NOT EXISTS index_name
ON keyspace.table_name ( column_name )
```

And then run the `describe` on `movies` that shows `CREATE INDEX` listed in the output.

Take a screenshot of the command you used and the output.

```
cassandra@cqlsh:entertainment> describe movies
CREATE TABLE entertainment.movies (
    id text PRIMARY KEY,
    director text,
    rating text,
    title text,
    year text
) WITH additional_write_policy = '99p'
AND bloom_filter_fp_chance = 0.01
AND caching = 'keys', 'rows_per_partition': 'NONE'
AND cdc = false
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '16'}
AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND memtable = 'default'
AND max_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND max_index_interval = 2000
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair = 'BLOCKING'
AND speculative_retry = '99p'

CREATE INDEX rating_index ON entertainment.movies (rating);
```

Assessment: Take a screen capture of the output and save the screen capture as 09-movies-rating-index.png (You can save the image using either the .jpg or .png extension).

Task 10 - Write a cql query to count the number of movies that are rated 'G'.

▼ Click here for Hint

use the `SELECT COUNT(*) FROM table_name WHERE criteria` command

```
SELECT COUNT(*) FROM table_name WHERE some_field='some value';
```

```
cassandra@cqlsh:entertainment> SELECT COUNT(*) FROM movies WHERE rating='G';
count
-----
 32
(1 rows)
Warnings :
Aggregation query used without partition key
```

Assessment: Take a screen capture of the output and save the screen capture as 10-g-rated-movies.png (You can save the image using either the .jpg or .png extension).

Checklist for submission

1. **Assessment:** Take a screen capture of the output and save the screen capture as 01-mongo-import.png.
2. **Assessment:** Take a screen capture of the output and save the screen capture as 02-most-movies-year.png.
3. **Assessment:** Take a screen capture of the output and save the screen capture as 03-movies-count-1999.png.
4. **Assessment:** Take a screen capture of the output and save the screen capture as 04-average-votes.png.
5. **Assessment:** Take a screen capture of the output and save the screen capture as 05-mongo-export.png.
6. **Assessment:** Take a screen capture of the output and save the screen capture as 06-describe-keyspaces.png.
7. **Assessment:** Take a screen capture of the output and save the screen capture as 07-movies-imported.png.
8. **Assessment:** Take a screen capture of the output and save the screen capture as 08-movies-count.png.
9. **Assessment:** Take a screen capture of the output and save the screen capture as 09-movies-rating-index.png.
10. **Assessment:** Take a screen capture of the output and save the screen capture as 10-g-rated-movies.png.

Congratulations! That's a wrap!

Author: [Muhammad Yahya](#)

© IBM Corporation. All rights reserved.