Report

Assignment 3 - MongoDB

Group: 22
Students: Henrik Moe Wæhre, Mari Lehne, Dominika Iza Kowalska

Introduction

In this assignment, we used MongoDB to solve different database tasks based on the trajectories dataset included in the assignment. The tasks were divided into two parts; first, was cleaning and inserting data into the database tables, and second was to write queries to the database.

In the first part of the assignment the group decided that working together physically on campus would be best to solve the assignment since it was hard to assign tasks evenly . This way, every member of the group had an overview of all done in the assignment. In addition, we used the GitHub repository to push the code so that the whole group could access the assignment on their computers. In the second part of the assignment the queries tasks were evenly distributed between group members to work on outside meeting hours.

Link to the GitHub repository: https://github.com/henrikmw/TDT4225_exercise_3

We created a .env file in order to connect to the database. The .env file contains:
DBHOST=tdt4225-22.idi.ntnu.no
DBNAME=db_group22
DBUSER=testuser
DBPASS=test123

Results

Part 1:

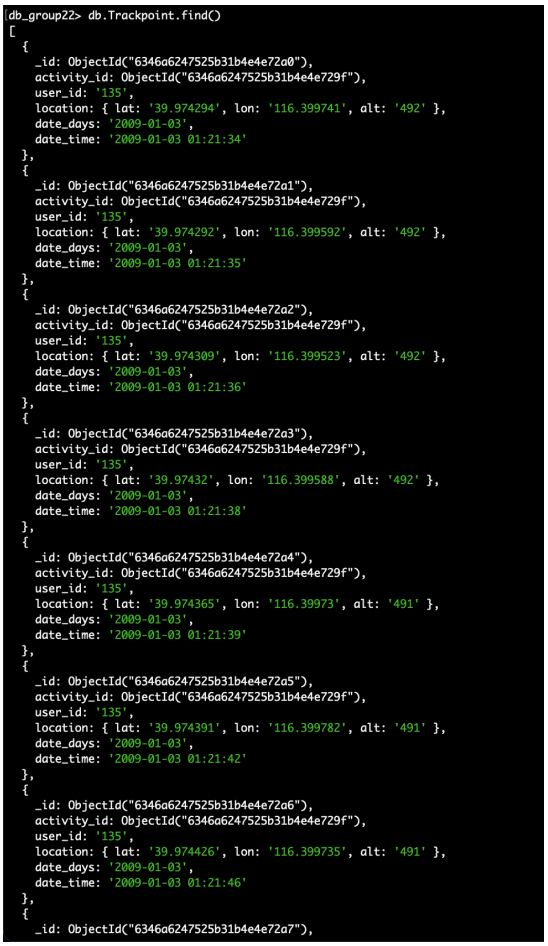
User Table:

```
db_group22> db.User.find()
[
  {
    _id: '000',
    has_labels: 0,
    activities: [
      ObjectId("63514b0c9cf59a40c55535fa"),
      ObjectId("63514b0c9cf59a40c5553cd5"),
      ObjectId("63514b0c9cf59a40c5554007"),
      ObjectId("63514b0c9cf59a40c55540bf"),
      ObjectId("63514b0c9cf59a40c5554808"),
      ObjectId("63514b0c9cf59a40c5554882"),
      ObjectId("63514b0c9cf59a40c5555045"),
      ObjectId("63514b0c9cf59a40c55550f3"),
      ObjectId("63514b0c9cf59a40c55557f0"),
      ObjectId("63514b0c9cf59a40c5555a77"),
      ObjectId("63514b0c9cf59a40c5555b83"),
      ObjectId("63514b0c9cf59a40c5555c1c"),
      ObjectId("63514b0c9cf59a40c5555e0b"),
      ObjectId("63514b0c9cf59a40c55562b1"),
      ObjectId("63514b0c9cf59a40c555661d"),
      ObjectId("63514b0c9cf59a40c5556735"),
      ObjectId("63514b0c9cf59a40c55567ef"),
      ObjectId("63514b0c9cf59a40c555681b"),
      ObjectId("63514b0c9cf59a40c5556a7e"),
      ObjectId("63514b0c9cf59a40c5557029"),
      ObjectId("63514b0c9cf59a40c55570bb"),
      ObjectId("63514b0c9cf59a40c5557283"),
      ObjectId("63514b0c9cf59a40c5557309"),
      ObjectId("63514b0c9cf59a40c5557381"),
      ObjectId("63514b0c9cf59a40c55573f4"),
      ObjectId("63514b0c9cf59a40c55574d3"),
      ObjectId("63514b0c9cf59a40c5557ddc"),
      ObjectId("63514b0c9cf59a40c555841a"),
      ObjectId("63514b0c9cf59a40c5558591"),
      ObjectId("63514b0c9cf59a40c55585ad"),
      ObjectId("63514b0c9cf59a40c55585fd"),
      ObjectId("63514b0c9cf59a40c5558653"),
      ObjectId("63514b0c9cf59a40c5558de4"),
      ObjectId("63514b0c9cf59a40c5558dfe"),
      ObjectId("63514b0c9cf59a40c555901b"),
      ObjectId("63514b0c9cf59a40c5559080"),
      ObjectId("63514b0c9cf59a40c55590eb"),
      ObjectId("63514b0c9cf59a40c55590f3"),
      ObjectId("63514b0c9cf59a40c5559114"),
      ObjectId("63514b0c9cf59a40c55596da"),
      ObjectId("63514b0c9cf59a40c55597a4"),
      ObjectId("63514b0c9cf59a40c555984d"),
      ObjectId("63514b0c9cf59a40c55598ac"),
      ObjectId("63514b0c9cf59a40c555a1be"),
      ObjectId("63514b0c9cf59a40c555a633"),
      ObjectId("63514b0c9cf59a40c555ad3f"),
      ObjectId("63514b0c9cf59a40c555adfa"),
      ObjectId("63514b0c9cf59a40c555ae0d"),
      ObjectId("63514b0c9cf59a40c555b472"),
      ObjectId("63514b0c9cf59a40c555b6f3"),
      ObjectId("63514b0c9cf59a40c555baf1"),
      ObjectId("63514b0c9cf59a40c555bbf0"),
      ObjectId("63514b0c9cf59a40c555bc42"),
      ObjectId("63514b0c9cf59a40c555beec"),
```

Activity table:

```
db_group22> db.Activity.find()
[
  {
    _id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    transportation_mode: 'NULL',
    start_date_time: '2009/01/03 01:21:34',
    end_date_time: '2009/01/03 05:40:31',
    trackpoints: [
      ObjectId("6346a6247525b31b4e4e72a0"),
      ObjectId("6346a6247525b31b4e4e72a1"),
      ObjectId("6346a6247525b31b4e4e72a2"),
      ObjectId("6346a6247525b31b4e4e72a3"),
      ObjectId("6346a6247525b31b4e4e72a4"),
      ObjectId("6346a6247525b31b4e4e72a5"),
      ObjectId("6346a6247525b31b4e4e72a6"),
      ObjectId("6346a6247525b31b4e4e72a7"),
      ObjectId("6346a6247525b31b4e4e72a8"),
      ObjectId("6346a6247525b31b4e4e72a9"),
      ObjectId("6346a6247525b31b4e4e72aa"),
      ObjectId("6346a6247525b31b4e4e72ab"),
      ObjectId("6346a6247525b31b4e4e72ac"),
      ObjectId("6346a6247525b31b4e4e72ad"),
      ObjectId("6346a6247525b31b4e4e72ae"),
      ObjectId("6346a6247525b31b4e4e72af"),
      ObjectId("6346a6247525b31b4e4e72b0"),
      ObjectId("6346a6247525b31b4e4e72b1"),
      ObjectId("6346a6247525b31b4e4e72b2"),
      ObjectId("6346a6247525b31b4e4e72b3"),
      ObjectId("6346a6247525b31b4e4e72b4"),
      ObjectId("6346a6247525b31b4e4e72b5"),
      ObjectId("6346a6247525b31b4e4e72b6"),
      ObjectId("6346a6247525b31b4e4e72b7"),
      ObjectId("6346a6247525b31b4e4e72b8"),
      ObjectId("6346a6247525b31b4e4e72b9"),
      ObjectId("6346a6247525b31b4e4e72ba"),
      ObjectId("6346a6247525b31b4e4e72bb"),
      ObjectId("6346a6247525b31b4e4e72bc"),
      ObjectId("6346a6247525b31b4e4e72bd"),
      ObjectId("6346a6247525b31b4e4e72be"),
      ObjectId("6346a6247525b31b4e4e72bf"),
      ObjectId("6346a6247525b31b4e4e72c0"),
      ObjectId("6346a6247525b31b4e4e72c1"),
      ObjectId("6346a6247525b31b4e4e72c2"),
      ObjectId("6346a6247525b31b4e4e72c3"),
      ObjectId("6346a6247525b31b4e4e72c4"),
      ObjectId("6346a6247525b31b4e4e72c5"),
      ObjectId("6346a6247525b31b4e4e72c6"),
      ObjectId("6346a6247525b31b4e4e72c7"),
      ObjectId("6346a6247525b31b4e4e72c8"),
      ObjectId("6346a6247525b31b4e4e72c9"),
      ObjectId("6346a6247525b31b4e4e72ca"),
      ObjectId("6346a6247525b31b4e4e72cb"),
      ObjectId("6346a6247525b31b4e4e72cc"),
      ObjectId("6346a6247525b31b4e4e72cd"),
      ObjectId("6346a6247525b31b4e4e72ce"),
      ObjectId("6346a6247525b31b4e4e72cf"),
      ObjectId("6346a6247525b31b4e4e72d0"),
      ObjectId("6346a6247525b31b4e4e72d1"),
      ObjectId("6346a6247525b31b4e4e72d2"),
```

Trackpoint table:



```
db_group22> db.Trackpoint.find()
[
  {
    _id: ObjectId("6346a6247525b31b4e4e72a0"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.974294', lon: '116.399741', alt: '492' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:34'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a1"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.974292', lon: '116.399592', alt: '492' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:35'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a2"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.974309', lon: '116.399523', alt: '492' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:36'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a3"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.97432', lon: '116.399588', alt: '492' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:38'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a4"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.974365', lon: '116.39973', alt: '491' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:39'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a5"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.974391', lon: '116.399782', alt: '491' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:42'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a6"),
    activity_id: ObjectId("6346a6247525b31b4e4e729f"),
    user_id: '135',
    location: { lat: '39.974426', lon: '116.399735', alt: '491' },
    date_days: '2009-01-03',
    date_time: '2009-01-03 01:21:46'
  },
  {
    _id: ObjectId("6346a6247525b31b4e4e72a7"),
```

Part 2:

1. How many users, activities and trackpoints are there in the dataset (after it is inserted into the database).

```
User has 182 documents
Activity has 7877 documents
Trackpoint has 9681756 documents
```

2. Find the average number of activities per user.

```
Average number of activities per user: 51.82236842105263
```

3. Find the top 20 users with the highest number of activities.

```
List of top 20 user with highest number of activities
1. User 025: 715 activities
2. User 128: 519 activities
3. User 062: 406 activities
4. User 041: 399 activities
5. User 004: 346 activities
6. User 140: 345 activities
7. User 017: 265 activities
8. User 003: 261 activities
9. User 014: 236 activities
10. User 030: 210 activities
11. User 011: 201 activities
12. User 039: 198 activities
13. User 034: 180 activities
14. User 000: 155 activities
15. User 002: 146 activities
16. User 142: 138 activities
17. User 037: 129 activities
18. User 013: 119 activities
19. User 042: 110 activities
20. User 020: 94 activities
```

4. Find all users who have taken a taxi.

```
Users that have taken taxi:
085
078
062
098
111
128
163
080
010
058
```

5. Find all types of transportation modes and count how many activities that are tagged with these transportation mode labels. Do not count the rows where the mode is null.

```
airplane has count 3
bike has count 262
boat has count 1
bus has count 199
car has count 419
run has count 1
subway has count 133
taxi has count 37
train has count 2
walk has count 481
```

6.

a. Find the year with the most activities.
b. Is this also the year with most recorded hours?

```
TASK 6a

2009 had the most activities

TASK 6b

The year with most hours is 2009 at 8906 hours
This is the same year as the one with the most activities
```

7. Find the total distance (in km) walked in 2008, by user with id=112.

```
User 112 walked 3630.331531941775 km in 2008
```

8. Find the top 20 users who have gained the most altitude meters.

User ID	Altitude gained
035	71464
017	63493.9
123	57364.1
122	42715.6
012	40306.1
130	31504.6
066	31503
002	29862.2
028	29380.9
054	28685.2
027	27944.1
007	27669.7
040	26341.4
013	26312.8
064	26088.1
121	24041.9
145	23932.6
062	23477.5
135	23462
070	23061.2

9. Find all users who have invalid activities, and the number of invalid activities per user.

The result was too long to screenshot.
000 has 101 invalid activities
001 has 45 invalid activities
002 has 98 invalid activities
003 has 179 invalid activities
004 has 219 invalid activities
005 has 45 invalid activities
006 has 17 invalid activities
007 has 30 invalid activities
008 has 16 invalid activities
009 has 31 invalid activities
011 has 32 invalid activities
012 has 43 invalid activities
013 has 29 invalid activities
014 has 118 invalid activities
015 has 46 invalid activities
016 has 20 invalid activities
017 has 129 invalid activities
018 has 27 invalid activities
019 has 31 invalid activities
020 has 11 invalid activities
021 has 1 invalid activities
022 has 55 invalid activities
023 has 11 invalid activities
024 has 27 invalid activities
025 has 263 invalid activities
026 has 18 invalid activities
027 has 2 invalid activities
028 has 36 invalid activities
029 has 25 invalid activities
030 has 112 invalid activities
031 has 3 invalid activities
032 has 12 invalid activities
033 has 2 invalid activities
034 has 88 invalid activities
035 has 23 invalid activities
036 has 34 invalid activities
037 has 100 invalid activities
038 has 58 invalid activities
039 has 147 invalid activities
040 has 17 invalid activities

041 has 201 invalid activities
042 has 55 invalid activities
043 has 21 invalid activities
044 has 32 invalid activities
045 has 7 invalid activities
046 has 13 invalid activities
047 has 6 invalid activities
048 has 1 invalid activities
050 has 8 invalid activities
051 has 36 invalid activities
054 has 2 invalid activities
055 has 15 invalid activities
056 has 1 invalid activities
057 has 16 invalid activities
058 has 2 invalid activities
060 has 1 invalid activities
061 has 12 invalid activities
062 has 134 invalid activities
063 has 8 invalid activities
065 has 5 invalid activities
066 has 6 invalid activities
067 has 1 invalid activities
069 has 1 invalid activities
070 has 5 invalid activities
071 has 29 invalid activities
072 has 2 invalid activities
073 has 6 invalid activities
074 has 19 invalid activities
075 has 1 invalid activities
076 has 2 invalid activities
077 has 3 invalid activities
078 has 5 invalid activities
079 has 2 invalid activities
080 has 1 invalid activities
081 has 2 invalid activities
083 has 15 invalid activities
084 has 4 invalid activities
085 has 11 invalid activities
086 has 2 invalid activities
087 has 1 invalid activities
089 has 4 invalid activities

090 has 3 invalid activities
092 has 1 invalid activities
093 has 4 invalid activities
094 has 16 invalid activities
095 has 4 invalid activities
097 has 2 invalid activities
099 has 11 invalid activities
103 has 24 invalid activities
108 has 2 invalid activities
109 has 3 invalid activities
111 has 1 invalid activities
112 has 11 invalid activities
113 has 1 invalid activities
115 has 26 invalid activities
119 has 22 invalid activities
121 has 4 invalid activities
122 has 6 invalid activities
123 has 3 invalid activities
126 has 3 invalid activities
127 has 4 invalid activities
128 has 139 invalid activities
130 has 8 invalid activities
131 has 10 invalid activities
132 has 3 invalid activities
133 has 4 invalid activities
134 has 31 invalid activities
135 has 5 invalid activities
139 has 2 invalid activities
140 has 86 invalid activities
142 has 52 invalid activities
144 has 1 invalid activities
145 has 5 invalid activities
146 has 7 invalid activities
150 has 16 invalid activities
151 has 1 invalid activities
152 has 2 invalid activities
153 has 2 invalid activities
155 has 30 invalid activities
157 has 9 invalid activities
158 has 9 invalid activities
159 has 5 invalid activities

162 has 9 invalid activities
163 has 4 invalid activities
164 has 6 invalid activities
165 has 2 invalid activities
166 has 2 invalid activities
167 has 4 invalid activities
168 has 19 invalid activities
169 has 9 invalid activities
171 has 3 invalid activities
172 has 9 invalid activities
173 has 5 invalid activities
175 has 1 invalid activities
176 has 8 invalid activities
180 has 2 invalid activities
181 has 14 invalid activities

10. Find the users who have tracked an activity in the Forbidden City of Beijing.

```
TASK 10  ---------------------------------
User 004 has tracked activity inside the forbidden city of Beijing
User 018 has tracked activity inside the forbidden city of Beijing
User 019 has tracked activity inside the forbidden city of Beijing
User 131 has tracked activity inside the forbidden city of Beijing
```

11. Find all users who have registered transportation_mode and their most used transportation_mode.

```
TASK 11 ---------------------------------
User: 010  Most used transportation: taxi
User: 020  Most used transportation: bike
User: 021  Most used transportation: walk
User: 052  Most used transportation: bus
User: 056  Most used transportation: bike
User: 058  Most used transportation: taxi
User: 060  Most used transportation: walk
User: 062  Most used transportation: bus
User: 064  Most used transportation: bike
User: 065  Most used transportation: bike
User: 067  Most used transportation: walk
User: 069  Most used transportation: bike
User: 073  Most used transportation: walk
User: 075  Most used transportation: walk
User: 076  Most used transportation: car
User: 078  Most used transportation: walk
User: 080  Most used transportation: bike
User: 081  Most used transportation: bike
User: 082  Most used transportation: walk
User: 084  Most used transportation: walk
User: 085  Most used transportation: walk
User: 086  Most used transportation: car
User: 087  Most used transportation: walk
User: 089  Most used transportation: car
User: 091  Most used transportation: bus
User: 092  Most used transportation: walk
User: 097  Most used transportation: bike
User: 098  Most used transportation: taxi
User: 101  Most used transportation: car
User: 102  Most used transportation: bike
User: 107  Most used transportation: walk
User: 108  Most used transportation: walk
User: 111  Most used transportation: taxi
User: 112  Most used transportation: walk
User: 115  Most used transportation: car
User: 117  Most used transportation: walk
User: 125  Most used transportation: bike
User: 126  Most used transportation: bike
User: 128  Most used transportation: car
User: 136  Most used transportation: walk
User: 138  Most used transportation: bike
User: 139  Most used transportation: bike
User: 144  Most used transportation: walk
User: 153  Most used transportation: walk
User: 161  Most used transportation: walk
User: 163  Most used transportation: bike
User: 167  Most used transportation: bike
User: 175  Most used transportation: bus
```

Discussion

After reading the three texts recommended in the assignment description, we decided to use an array of references to the N-side objects for the database structure. We thought that it would suit the purpose of the assignment best since the collection has one-to-many relationships.

We learnt how to correctly set up a MongoDB database with user rights in part 1 of the assignment by following the setup instructions in the problem text. Furthermore, we learned how to connect the MongoDB server to the Ubuntu virtual machine. In addition, we learned how to insert data from a dataset into a MongoDB database.

During data cleaning in the first part of the assignment we have discovered that some users have multiple activities at the same time and different transportation modes. Since it is not possible to have multiple activities at the exact same time we have chosen to include only one of those activities.

When inserting data we had the same problem with long inserting time as in the previous assignment. We solved this problem by iterating through trackpoints in the Trajectory directory and checking each trackpoint against every line in labels.txt, instead of iterating all trackpoints in the Trajectory directory for every line in labels.txt. This reduces the inserting time significantly.

In part 2 of the assignment we learned multiple methods to select/match data from MongoDB using Python. When working on some of the queries we had problems with matching what we searched for. Even if the coded query should work there wast any result or the result was not correct. To solve this problem we had to change the User collection from being a dictionary to a list to make it possible to finish some of the queries. Some of the other queries done before needed to be redone because of the change.

MySQL vs MongoDB

Discuss the differences between MySQL and MongoDB (relational databases vs NoSQL databases).

There are many differences between MySQL and MongoDB, and here we will address the main differences. One of the main differences is that MySQL supports one-to-many relationships, butMongoDB does not. This is where the terms relational and non-relational come in. Another difference is that MySQL requires you to use predefined schemas to determine the structure of your data before you work with it, but a NoSQL database has dynamic schema for unstructured data.

Further, MySQL is vertically scalable, which means that you can increase the load on a single server by increasing things like RAM, CPU or SSD. On the other hand, NoSQL is horizontally scalable, meaning that you can handle more traffic by sharding, or adding more servers in your NoSQL database. Thus NoSQL can ultimately become larger and more powerful, making these databases the preferred choice for large or ever-changing data sets.

The last difference we want to mention is that SQL databases are table-based, which means it stores and organizes data in columns and rows as defined during table creation. The NoSQL databases are either key-value pairs, document-based, graph databases or wide-column stores.

Which database did you prefer to solve these tasks?
We preferred to solve these tasks with MongoDB. We experienced MongoDB to be much easier to use in terms of inserting data and how to handle the data. It was easier to insert data as there were no rules as to which collections to insert before others, due to MongoDB being non-relational. We also found it easier to do queries with MongoDB, which can be explained with us having previous experience with this and not with MySQL.

What are the pros and cons using one versus the other?

- **MySQL:**
  - **Pros:**
    - Relational SQL databases a better option for applications that require multi-row transactions such as an accounting system or for legacy systems
  - **Cons:**
    - Unsuited for large datasets and big data analysis
    - Requires the use of predefined schemas to determine the structure of data before you work with it and changing the structure can be quite confusing
    -
- **NoSQL:**
  - **Pros:**
    - Documents can be created without having a defined structure and so each document can have its own unique structure
    - Good to use when having large datasets and for big data analysis because it is horizontally scalable
    - Suited for hierarchical data storage.
  - **Cons:**
    - Not so good for complex queries