

MongoDB practical session

Ecole Centrale, January 2025, Előd Egyed-Zsigmond

Instructions

Rename the file: `MongoLab_NAME1_NAME2.txt.txt` which is in the archive containing the subject of the practical work by replacing NAME1 and NAME2 with your names. Insert your answers after the corresponding statement in the file. For the simple queries, insert the first lines (do not exceed thirty) of the result.

You must send your reports as an attachment by email to the address: elod.egyed-zsigmond@insa-lyon.fr before Tuesday, January 14th, 2025, 2:00 p.m.

Lab tools :

- **MongoDB server** Community, v8.x (download the package zip on Windows or tgz on MacOS/Linux) : <https://www.mongodb.com/try/download/community> .
- **MongoDB Shell** (a command-line tool for connecting to the database): Download the ZIP package for Windows or the TGZ package for macOS/Linux from <https://www.mongodb.com/try/download/shell> .
- **MongoDB Database Tools**: Used for loading data (download the ZIP version), available from <https://www.mongodb.com/try/download/database-tools>.
- A **code editor** for writing non-autocorrecting code (e.g., Notepad++, Sublime Text, VS Code, Emacs, Notepad, Eclipse, etc.). Avoid text editors like MS Word, LibreOffice, or Google Docs that may modify your text without informing you.
- The **MongoDB client**: Studio 3T Free, downloadable from <https://studio3t.com/download-studio3t-free>.

Preparation

A. Installing and Launching the MongoDB Server

Install the MongoDB Server

- Download the MongoDB server from the link provided earlier (zip archive).
- Note the installation directory (referred to as `MongoDBServerHome`).
- Unzip the server archive to the `MongoDBServerHome`
- Eventually refer to the [official installation documentation](#) for detailed steps.

Create a Data Directory

- Set up a directory to store MongoDB data. Typically, use `C:\data\db`.
- This directory is referred to as `MongoDBDataDir`.

Start the MongoDB Server from the Command Line

- Detailed instructions are available in the [official guide](#).

Steps to Run MongoDB:

a. On Windows, open a Command Prompt:

- Press `Windows + R`, type `cmd`, and press `Enter`.

b. Navigate to the `bin` directory inside the MongoDB installation folder (`MongoDBServerHome\bin`) by using the `cd` command:

```
cd <MongoDBServerHome>\bin
```

(Replace <MongoDBServerHome> with the actual installation directory path.)

c. Start the MongoDB server by typing:

```
mongod --dbpath "MongoDBDataDir"
```

(Replace "MongoDBDataDir" with the path to your MongoDB data directory created in step 2.)

d. JSON-formatted messages will appear in the Command Prompt. The last message should end with something like:

```
... "ctx": "listener", "msg": "Waiting for  
connections", "attr": { "port": 27017, "ssl": "off" } }
```

e. The MongoDB server is now running locally and is listening for connections on port **27017**.

Connecting to the MongoDB Server and Creating the Database for Practice

Steps to Connect and Create the Database

Install the MongoDB client

- Download the MongoDB client from the link provided earlier (zip archive).
- Unzip the client to a directory noted `MongoDBClientHome`

Open a New Command Prompt

- Press `Windows + R`, type `cmd`, and press `Enter`.

Navigate to the MongoDB `bin` Directory

- Unzip the MongoDB zip package In the Command Prompt, move to the `bin` directory inside the MongoDB installation folder (`MongoDBClientHome\bin`) using the `cd` command:
- `cd < MongoDBClientHome > \bin`

(Replace < MongoDBClientHome > with the actual installation directory path.)

Launch the MongoDB Client

- Start the MongoDB client by typing:
- `mongosh`
- After a few messages, a prompt with the `>` symbol will appear, indicating that you are connected to the MongoDB server.

Create the Database

- At the `>` prompt, create the database for the practice session by typing:
- `use mongoLab`
- The output will confirm that you have switched to the database `mongoLab`. If the database does not already exist, this command will create it.

You are now connected to the MongoDB server and have successfully created the `mongoLab` database.

Populating the Database

Steps to Import Data into the MongoDB Database

1. Unzip MongoDB Database Tools

- Extract the ZIP file containing the MongoDB Database Tools.

2. Move the `mongoimport` Executable

- Locate the `mongoimport.exe` file (or its equivalent for your OS) in the unzipped tools folder.
- Move it to the `MongoDBServerHome\bin` directory, where `mongod.exe` is located.

3. Copy the Data File

- Copy the `velov_geo.json` file from the TP archive into the `MongoDBServerHome\bin` directory.
- This file contains JSON documents describing the Vélo’V stations. *(These data were available on the [Grand Lyon data site](#) on January 11, 2021, under [this dataset](#).)*

4. Open a New Command Prompt

- Press `Windows + R`, type `cmd`, and press `Enter`.
- Navigate to the `MongoDBServerHome\bin` directory:
- `cd <MongoDBServerHome>\bin`

5. Ensure Required Files Are in Place

- Verify that both the `mongoimport` executable and the `velov_geo.json` file are in the `MongoDBServerHome\bin` directory.

6. Run the Data Import Command

- From the Command Prompt (not the MongoDB client), execute the following command to import the data:

```
mongoimport --db mongoLab --collection velov_geo --file
"velov_geo.json"
```

- This command will:
 - Create the `mongoLab` database if it does not already exist.
 - Create the `velov_geo` collection.
 - Insert the documents from the `velov_geo.json` file into the collection.
- **Note:**

- You can run `mongoimport` from any location if its path is added to your system's `PATH` environment variable.
- Ensure the `--file` parameter points to the full path of the JSON file if it is not in the current directory.

7. Verify the Data Import

- Open the MongoDB client (`mongosh`) as described earlier.
- Switch to the `mongoLab` database:
 - `use mongoLab`
- List the imported data using the command:
 - `db.velov_geo.find().pretty()`
- You should see JSON documents with Vélo’V station information displayed, along with the message:
 - Type `"it"` for more

8. Database Creation and Population Complete

- Your `mongoLab` database is now fully created and populated with the `velov_geo` collection containing the Vélo’V station data.

Simple Queries

Express simple queries for the following searches:

1. List all stations located in the city of **VILLEURBANNE**.
2. List all stations **not** located in **VILLEURBANNE**.
3. List all stations that do not have a second address (**address2** is empty).
4. Count the number of stations where the second address has a value.
5. List stations with more than 2 bikes available (**available_bikes**).
6. List all distinct cities (**communes**).
7. List all distinct cities, sorted in alphabetical order.
8. List stations in the 9th arrondissement (**Lyon 9ème**) in ascending order of available bikes.
9. Project results onto the station name, address, and number of available bikes.
10. Display cities and the number of Vélo’V stations in each commune.
11. Sort the results by commune.
12. Sort the results by the number of stations in descending order.
13. Count the number of Vélo’V stations in **VILLEURBANNE**, grouped by the number of available bikes, sorted by this number.
14. Add to the previous result the names of stations for each number of available bikes.
15. Calculate the average number of available bikes per city.

Geographical Queries

Create a 2D Sphere Index

1. **Specify the Coordinates as Geographical Data**
 - o Use the following commands to create a 2D sphere index:
 - o `use mongoLab;`
 - o `db.velov_geo.createIndex({ "geometry.coordinates": "2dsphere" });`
2. **Verify the Index Creation**
 - o Confirm that the index has been created with the command:
 - o `db.velov_geo.getIndexes();`

Geographical Queries

16. Find Vélo’V Stations Within 500m of a Point

- Look for Vélo’V stations within a radius of 500 meters from the point:

```
$geometry: {
  type: "Point",
  coordinates: [ 4.863132722360224, 45.77022676914935 ]
}
```

(Use the \$near operator.)

17. List the 5 Closest Stations to Specific Coordinates

- List the five nearest stations to the coordinates:

```
{
```

```

    type: "Point",
    coordinates: [ 4.863132722360224, 45.77022676914935 ]
  }

```

(Use the \$geoNear operator.)

Textual Queries

Create a Text Index

1. Specify the Field as Text

- Create a text index on the field `properties.pole`:

```

db.velov_geo.createIndex({ "properties.pole": "text" }, {
  default_language: "french" });

```

Textual Query

18. Find Stations Where the `pole` Field Matches "quartiers"

- Search for stations where the `pole` field responds to the query "quartiers":

```

db.velov_geo.find({ $text: { $search: "quartiers" } }).pretty();

```

19 Adding Two New Vélo’V Stations to the Database

Steps:

a) Find Information About Two Stations

- Visit the [Velov data page at the grandlyon website](#).
- Locate two stations and note down their details, such as:
 - Station name
 - Location (latitude and longitude)
 - Total number of stands
 - Number of available bikes
 - Number of free stands

b) Extract/create JSON Files for the Two Stations

- Using a text editor (e.g., Notepad++, Sublime Text, or VS Code), create a JSON file with the structure matching the existing `velov_geo.json` format.
- Describe your procedure and provide the 2 json elements

c) Add the two stations to the collection. Describe your procedure

20 Data enrichment

Look for downloadable data in JSON format on the website <https://data.grandlyon.com/> (or elsewhere) that can complement the information of Vélo’V stations.

- Provide the URL that allows you to download the data and explain how it can complement the MongoDB collection of Vélo’V stations.
- Download the data and import it into MongoDB. Describe your procedure.

- c) Provide at least 2 MongoDB queries that combine the new information with the old data. Describe your method. You can combine the data first and apply the queries on the combined collection.