UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Big Data Management (BDMA & MIRI Masters)
Session: Document Stores

Alberto Abelló & Sergi Nadal

In this session we will work with MongoDB, nowadays the most popular document store.

Instructions

All tools you need are provided in the BDM virtual machine, accessible via *Virtech*. Please, refer to the provided manual for details on how to set up the virtual machine. Notice that as lab sessions are performed in couples, only one virtual machine will be required during the session. The code for the lab session can be found in LearnSQL.

Once you have fetched the code, it can be imported as a *Maven Project* into eclipse. All executions can be run locally, and thus you do not need to be concerned with the host connection. In the *Main* class you can see that different arguments are expected, and will drive the behavior of the program. The simplest way to define such arguments is by creating a *Run Configuration* of type *Java Application* from the *Run* menu. In the tab menu called *Arguments* you will be able to edit such arguments, in each exercise you will be guided with the required arguments.

Delivery

By the end of the session hand in your answers. Coding exercises must be delivered via LearnSQL. Prepare a ZIP file and upload it to LearnSQL, with the following content: (1) `populateDB.java`, and (2) `queries.txt`. **The system will close the submission five minutes before the end of the class (unless said otherwise by the lab professor), make sure to submit your file before!**

Design a database and a workload

Given the UML conceptual diagram depicted in Figure 1, which depicts five classes with different attributes (note primary keys are underlined), you are asked to perform the following tasks:

1. Model it in terms of collections and documents according to the workload described in (2).
2. Implement the following queries:
   - Q_1: Average person age per country code
   - Q_2: Average person age per city name
   - Q_3: Number of employees per company
   - Q_4: Number of employees per country
3. Justify the proposed data model.

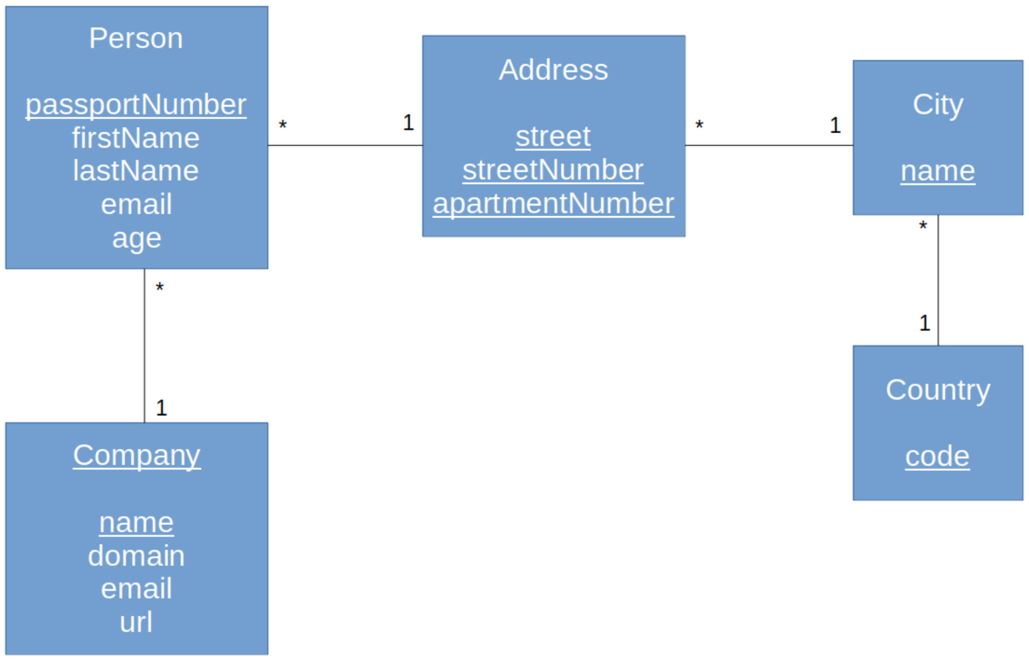

Figure 1: UML conceptual diagram

Exercise 1: Model your database (25% weight)

Here, you must design the database in MongoDB (i.e., in terms of documents) in such a way the performance of the four presented queries is optimal. Do not use indexing structures, focus on the structure of the documents within the different collections. You might assume the workload follows an equiprobable distribution. If you need to make any assumption, make it explicit in Exercise 3.

Implement a Java function to populate your database with N instances of the class *Person* (together with all its associated elements).

Note: due to the randomness in the data generator, obtaining the same primary key twice might not mean the other attributes are the same. For instance, company name *Adapt* can have domain *adapt.com* first and then *adapt.biz*. In this data population task, you can ignore such updates even though it might lead to inconsistencies.

Exercise 2: Implement the four queries (25% weight)

Here, we ask you to implement the four queries in the MongoDB shell. Provide their implementation in a file named `queries.txt`. Use the **Aggregation Framework** whenever possible.

Exercise 3: Rationale (50% weight)

Discuss all design decisions that you have made in the provided sheet. Be concise. Provide a sample JSON for each of the proposed collections. Some aspects you should cover are:

- Why did you choose this modeling approach?
- What are the trade-offs you had to make?
- For each query, is it optimal? Why/why not?
- What is the impact of updates in the chosen model?