Barcelona School of Economics

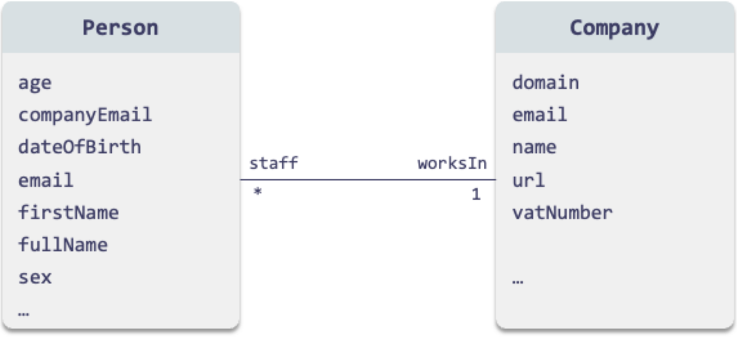
23D020: Big Data Management for Data Science
Lab 1: Document Stores

Assignment

Note: This is a hands-on lab on Document Stores. We will be using one of the most popular document databases: `MongoDB`*. We will practice how to import, create and model document databases, as well as how to query them. In the training document, we provide instructions for setting up the environment and here we list the exercises to be solved. One group member, in the name of the group, must upload the solution. Remember to include the name of all group members in your solutions. Please check the assignment deadline and be sure to meet it. It is a strict deadline!*

A Lab Statement

In this lab you will explore the different modeling alternatives in MongoDB. You will work with the following conceptual model depicted in UML.



The queries that you will need to implement with this model are the following:

Q1: For each person, retrieve their full name and their company's name.

Q2: For each company, retrieve its name and the number of employees.

Q3: For each person born before 1988, update their age to "30".

Q4: For each company, update its name to include the word "Company".

In this exercise you are asked to design the database using the three following models:

M1: Two types of documents, one for each class and referenced fields.

M2: One document for "Person" with "Company" as embedded document.

M3: One document for "Company" with "Person" as embedded documents.

B Python Implementation

For each design model, you need to **implement in Python** the following tasks:

1. Using `Faker` (a random Python data generator), generate random data for persons and companies. Be consistent with the number of companies and the proportion of employers for the three models. That is, use that the same number of companies and employees in the three models in order to make them comparable. The assumption is that you are modeling the same data with three different models.
2. Insert the data into MongoDB with each of the specified models.
3. Program queries Q1, Q2, Q3 and Q4 for each of the models and write their results in the console.
4. Measure the execution time of each query by adding the following instructions:

```
start_time = time.time()
/** Query code ... **/
query_time = end_time - time.time()
```

To aid you in doing the exercise, in the provided python project you can find sample code for the above tasks in the class `example.py`. For a full reference on MongoDB Python API you can check `https://docs.mongodb.com/drivers/pymongo/`.

C Results and Discussion

Once you have completed the above tasks, fill this table with the query execution times obtained with a high number of documents (e.g., at least 50000).

Table 1: Query Execution Times per Model

	Q1	**Q2**	**Q3**	**Q4**
M1				
M2				
M3				

Afterwards, answer the following questions (make sure you justify your answers and not only list them):

1. Order queries from best to worst for **Q1**. Which model performs best? Why?

2. Order queries from best to worst for **Q2**. Which model performs best? Why?

3. Order queries from worst to best for **Q3**. Which model performs **worst**? Why?

4. Order queries from worst to best for **Q4**. Which model performs **worst**? Why?

5. What are your conclusions about denormalization or normalization of data in MongoDB? In the case of updates, which offers better performance?

Deliverables

1. Python scripts to implement the tasks defined in Section B. The scripts must be included in a single zip file.

   - The Python code must include comments to facilitate the understanding. At the header of each file, include an overall comment explaining what are the steps implemented in the pipeline, and refer to these steps when explaining the code in the subsequent comments.
   - The execution of the three pipelines should be facilitated. For instance, the code should not include absolute paths or fixed user credentials (e.g., they should be requested by the user or stored in configuration files).

2. A PDF file (max two A4 pages) to answer the questions in Section C. In this document you can also include any assumptions made or justify the decisions you made (if any).

Assessment Criteria

i) Conciseness of explanations

ii) Understandability

iii) Coherence

iv) Soundness