# Artificial Intelligence & Cloud

## Project Assignment - MongoDB

### Main Goals

This project consists in the creation of a MongoDB database containing IMDB Movie data. This project covers:

- Data ingestion and curation.
- Standard query resolution.
- Aggregation query resolution using MongoDB aggregation framework.
- Aggregation query resolution using Map-reduce paradigm.

### Data specifications

The data collection proposed for the project is related to a collection of movies' information. The collection can be downloaded from the "Portale della Didattica" (imdb.json). This collection includes information about movies and their associated reviews on IMDB and Rotten Tomatoes.

### Dataset ingestion and curation

The data collection is provided in JSON format and needs to be imported in MongoDB.
Identify at least two attributes that are not imported using the correct data type (e.g., `imdb.votes`) and provide the function to correctly parse the data type (e.g., strings to integer) for those attributes: update all documents accordingly.
Investigate what could be the consequences of incorrect data type ingestion in the analytic phase.

### Standard query resolution

Use MongoDB Compass to resolve the following query of interest:

1. Find all the movies which have been scored higher than 4.5 on Rotten Tomatoes. The reviews for Rotten Tomatoes are contained in the `tomatoes` nested document. Sort the results using the ascending order for the release date.

2. Find the movies that have been written by 3 writers and directed by 2 directors.

3. For the movies that belong to the "Drama" genre and belong to the USA country, show their plot, duration (`runtime`), and title. Order the results according to the descending duration.

4. Find the movies satisfying all the following conditions:
   - have been published between 1900 and 1910
   - have an imdb rating higher than 9.0
   - contain the `fullplot` attribute

   In the results, show the following values:
   - the publication year,
   - the length of the full plot in terms of number of characters.

Sort the results according to the ascending order of the IMDB rating.

## Aggregation queries

Use the MongoDB aggregation framework to address the following queries.

1. Find the average rating score on Rotten Tomatoes for each publication year.

2. For movies that include Italy as a country, get the average number of directors. Be sure to consider only the movies that contain the list of directors.

3. Considering only movies that:
   a. contain information about IMDB score ratings
   b. contain a number for IMDB score ratings (you can check it by using `$type`)

   compute, separately for each movie's genre:
   - the average published year, and
   - the maximum score on IMDB.

4. Count the number of movies directed by each director. Sort the results according to the descending order of the number of directed movies.

## Map-reduce framework

Use the Map-reduce framework in MongoDB to address the following queries.

1. Find the number of movies published for each year.

2. Group movies according to their number of writers. For each group, find the average number of words in the title.
   NB: Check in the map function if the `writers` attribute is defined (i.e., if it exists).

3. Count the number of movies available for each language (attribute `languages`).
   NB: Check in the map function if the `languages` attribute is defined (i.e., if it exists).
   NB$_2$: It is possible to emit multiple pairs for each document using iterators over an array.

## Project assignment

### Report format

Write a report of including the following information:

1. **Project overview**
2. **Data ingestion and curation:** should include a short description of the data curation step.
3. **Standard queries resolution**: write the queries and explain the main operators used for their resolution. Provide a brief comment for each query from the analytical point of view.
4. **Aggregation framework and map-reduce**: write the aggregations and map-reduce functions designed for this step. Choose one query and explain its possible application in a real-world scenario.
5. **Interest query**: write and solve one additional novel query, defined by you, and describe when and how it could be helpful in a real use-case.

### Constraints on the report

The report must comply with the following constraints:

- The report must be generated using the standard IEEE conference template, available in LATEX and Word format.
  You are strongly encouraged to use the LaTeX version (you can also use Overleaf).

- The report must follow the division in the listed sections.

- The report must be, at most, 4 pages long.

### Submission

You should submit a single file (zip format) to the "Portale della Didattica" under the Homework section ("Elaborati"). The compressed folder must contain:

- The report in PDF format.
- The code snippets used in the data curation phase (data_curation.txt)
- The text of the queries for:
  - Standard queries resolution (standard_queries.txt)
  - Aggregation framework (aggregation_queries.txt)
- The code snippets used in the Map-reduce phase (map_reduce.txt)

The file must be named "Project_2_MongoDB.zip".

**Deadline:** April 2nd, 2021, end of the day (23:59 CET)

In case of doubts or problems, please contact Moreno La Quatra (moreno.laquatra@polito.it)