Guide for Project Development (mandatory)

Database NoSQL 2022/2023

The development of a project is mandatory to overcome the exam.

- Choose a project from one of the suggestions provided below, or propose your own via email.
- All use cases (suggested or proposed) must receive a written agreement by the Professor.
- Multiple choice question exam and project discussion can occur in separate exam sessions.
- A project can be chosen by a group composed of maximum 2 persons.
- Each project must be original!
- A PDF report containing an introduction to the project, a discussion about the motivation of the technologies used and the development decisions.
- Each project chosen must be communicated to the Professor via email, together with a GitHub account username (if you don't have one, simply create it).
- Students will receive a GitHub link with a repository where to upload the source code and all the project materials. Upload must occur MAXIMUM 2 days before the exam.
- An email must be sent to the Professor after the project upload on GitHub is completed.

Project development guideline:

1) Implement the chosen case study with all the databases studied (MySQL, MongoDB, Cassandra, Redis, and Neo4j)
2) Create 4 datasets of increasing size (250.000, 500.000, 750.000, and 1.000.000 records) and insert them in the chosen databases with the same information content (the result of a query must be the same in all the databases). The dataset can be created randomly using a random data generator (e.g. Mockaroo, https://www.mockaroo.com/) or via programming language (e.g. Faker library of

Python, https://faker.readthedocs.io/en/master/). It is suggested to create multiple csv files (for different dataset size) and only adapt the import phase for different databases. Each decision must be motivated in the final PDF report.

3) Make a comparison between the databases with the same dataset by defining 4 queries with increasing degrees of complexity from the point of view of the number of entities involved and the selection filters.
4) Choose a programming language of your choice in order to interact with the databases to carry out data entry (where necessary) and query execution operations. The experiments must be automated. Each experiment (e.g. query run time test) must be run at least 30 times considering the mean value and 95% confidence intervals (a tutorial on how to plot confidence intervals in a histogram is available by clicking here).
5) The results must be saved in an electronic spreadsheet to be processed and graphed. The histograms must be made considering the response times expressed in milliseconds (msec). For each query, plot the results as the dataset size changes (25%, 50%, 75%, and 100% of the information content). Since many NoSQL DBMS solutions use caching mechanisms, with the same dataset size, consider separately the first execution time and the average value of the following 30 executions for a total of 31 tests. For each established query, two histograms must be created: one with the first execution times as the size of the dataset changes, one with the average execution times as the size of the dataset varies.
6) Demonstrate which database on equal hardware / software conditions has better query execution times.
7) Create a short report (max 15-20 pages) including the following sections:
    - Problem addressed (max 2 points)
    - DBMS solution considered (brief description of the main characteristics of the database used) (max 2 points)
    - Design (containing description of the data model used) (max 6 points)
    - Implementation (containing the code used for data entry and for the implementation of each query) (max 6 points)
    - Experiments (containing tables with the response times obtained and related histograms) (max 6 points)

- Conclusions (max 3 points)
- From 0 to 6 additional points will be given for the oral presentation of the project.

8) *Maximum 2 days before the exam the student must send to the teacher by email:
   1. the report;
   2. the spreadsheet including experimental results and histograms;
   3. a link to a OneDrive folder including the developed code and a dump of the shortest dataset (25% of the information content).

**The day of the exam the students must discuss the developed working code to teacher either using his/her laptop or using a PC of the computer science laboratory.

Suggested use cases:

1. Social Networking Platform: Develop a NoSQL database to develop a social networking platform, which allows students and faculty members to connect with each other. Store user profiles, posts, messages, and other related data.
2. Course Management System: Develop a NoSQL database to develop a course management system that allows professors to manage course content, assignments, and grades. Store student and course-related data.
3. Research Project Management: Develop a NoSQL database to develop a research project management system that allows students and faculty members to collaborate on research projects. Store project-related data, such as research papers, data sets, and results.
4. Library Management System: Develop a NoSQL database for a library management system that allows students to borrow and return books. Store information related to books, borrowers, and borrowing history.
5. Online Learning Platform: Develop a NoSQL database to develop an online learning platform that allows students to access course materials, take quizzes, and submit assignments. Store course-related data, such as videos, quizzes, and assignments.

6. Event Management System: Develop a NoSQL database to develop an event management system that allows students and faculty to organize and schedule events. Store information related to events, attendees, and event feedback.
7. Student Information System: Develop a NoSQL database to develop a student information system that allows faculty and administrators to manage student data. Store information related to student grades, attendance, and demographics.
8. Learning Management System: Develop a NoSQL database to develop a learning management system that allows students to access course content, communicate with their instructors, and track their progress. Store information related to courses, assignments, quizzes, and student progress.
9. Personalized Recommendation System: Develop a NoSQL database to develop a personalized recommendation system that recommends courses, books, or other resources to students based on their interests and past behavior. Store information related to student interests, preferences, and past behavior.
10. Alumni Relations Management System: Develop a NoSQL database to develop an alumni relations management system that allows universities to manage and track interactions with their alumni. Store information related to alumni demographics, past donations, and interactions with the university.
11. Course Evaluation System: Develop a NoSQL database to develop a course evaluation system that allows students to provide feedback on their courses. Store information related to course evaluations, student feedback, and instructor performance.
12. Campus Safety and Security System: Develop a NoSQL database to develop a campus safety and security system that monitors and records security incidents and provides real-time alerts to security personnel. Store information related to incidents, security personnel, and camera feeds.
13. Scholarship Management System: Develop a NoSQL database to develop a scholarship management system that allows students to apply for scholarships and universities to manage the application process. Store information related to scholarship applications, selection criteria, and applicant data.
14. Online Admissions System: Develop a NoSQL database to develop an online admissions system that streamlines the admissions process for prospective students.

Store information related to applicant data, transcripts, test scores, and application status.

15. Curriculum Management System: Develop a NoSQL database to develop a curriculum management system that allows universities to manage course catalogs, course scheduling, and curriculum mapping. Store information related to courses, course schedules, and curriculum requirements.
16. Learning Analytics Platform: Develop a NoSQL database to develop a learning analytics platform that collects and analyzes data on student behavior, performance, and engagement. Store information related to student activities, learning outcomes, and performance metrics.