

# COMP 306: Database Management Systems

## Guidelines for Group Project

In the group project, you will pick a sample application area of DBMSs and develop a DBMS-powered software project. Your project **must** use a relational (SQL-based) DBMS in its back end. The choice of front end is left to you, i.e., you may decide to have a web interface, a desktop app, a mobile app, or a combination of them depending on what is appropriate.

Here are some sample project ideas. New and innovative ideas are welcome. You may discuss the suitability of your project idea with the instructor. I encourage you to pick a topic that is interesting for you and develop a project that you will be proud of, since this project can eventually become a good addition to your CV/portfolio.

- Online food or grocery ordering app (e.g., mini-Yemeksepeti)
- E-commerce or online retail website (e.g., mini-Amazon)
- University management app (e.g., mini-KUSIS)
- Real estate or vehicle search website (mini-Sahibinden)
- Library management app (books, authors, customers, lending, ...)
- Supermarket management app (products, employees, customers, purchases, ...)
- Web app for management of university events and social clubs
- Hospital management system (patients, doctors, appointments, prescriptions, ...)
- Website to track sports or e-sports leagues (clubs, players, matches, ratings, ...)
- **Movie tracking and rating site (e.g., mini-IMDB)**
- **Cinema or game store management app (customers, movies/games, purchases, ...)**
- ...

For the last two topics marked in **red**: I no longer recommend these project ideas since they are picked too frequently by too many groups. When multiple groups work on the same project idea, they are implicitly competing with each other. It is better to work on an underexplored or novel project idea.

**Group Size.** Recommended group size is 4-6 students per group. More than 6 students per group is not allowed. Less than 4 students per group is allowed, but if you choose to do this, know ahead of time that the workload for each group member will be high. The same grading criteria apply (i.e., project requirements remain the same) regardless of your group size, e.g., whether you are a 2-person group or a 5-person group.

**Demo Day.** We are planning to have a “Demo Day” at the end of the semester (probably in the second week of final exams) via Zoom. Each group will meet with the course staff one by one and demonstrate their project. The course staff will ask questions about the group’s design and implementation. Live attendance to the Demo Day is mandatory.

**Submission.** You need to submit:

- All source code and data that are necessary to run your project on our computers.
- A brief project report to fulfill the grading criteria below.

**Grading Criteria.** The group project is 10% of your overall course grade. First, we will grade the group as one whole unit, according to the following criteria. Then, some adjustments for each individual member’s grade can be made based on the group’s interactions with the course staff and the group members’ internal evaluations of one another (TBA later).

Note that we typically **curve** the whole class based on how good groups' projects are in comparison to each other. A few top projects receive full grade, and grades keep decreasing as we have less and less impressive projects. Thus, please take the percentages and grading criteria below as general guidelines and try to produce a good project overall, rather than trying to maximize your grade using 1-2 criteria and ignoring the others.

- [2%] ER and relational database design: Your project should have an intuitive and reasonably complex database design with multiple tables. You should draw an entity-relationship (ER) diagram and then convert the ER diagram to the relational model to obtain tables that have appropriate primary keys, foreign keys, and other constraints.
  - FAQ: How complex should our DB design be?
  - Answer: Ultimately, it depends on your project. But aim to have a design with at least 4-5 entities and 3-4 relationships (ER diagram), 5-6 interconnected tables (in your relational model).
- [1%] Populating your database: You should populate your database with sufficient amount of real or realistic data. When we execute queries on your database, we should get a meaningful number of tuples as output. When you demo your project, there should be enough data so that we can observe meaningful outputs.
- [3%] Writing advanced SQL queries: You will probably need many simple SQL queries for your project to be functional. In addition, you must have **at least 5 sophisticated SQL queries**, such as nested queries, queries with GROUP BY – HAVING, etc. These queries must be **meaningfully integrated into your project**, i.e., it is not sufficient for them to just be written on paper.
- [4%] Working prototype: The working prototype of your system should have a professional and nice-looking interface (GUI). The front-end (GUI) and back-end (DB) should be properly connected.

Note that many of the grading criteria above are interconnected. For example, if you don't have a working prototype, your advanced SQL queries cannot be integrated into your system. If you didn't populate your database, your queries will return empty results (or only few tuples) which will affect you negatively when you demonstrate your working prototype. If you don't have a good schema design, your SQL queries won't be meaningful, etc. Therefore, again, you should attempt to produce a good project overall by keeping the above criteria in mind, rather than optimizing for one or more specific criteria.