

Exercise - Start of Your Database Project

The project is a prerequisite that must be fulfilled in order to be allowed to participate in the exam. It will be graded on a pass/fail basis.

Your task:

Develop a database with a small GUI and some transactions. You can choose the DBMS and how you implement the GUI on your own. You can work in the lab or on your own computer.

Deliverable:

Project documentation plus complete code, by email and on paper. You don't need to turn in anything before the final due date.

Due date of the project: June 19.

Team size: two or three people

You may choose your teammate(s) as you wish, with the following restriction: Every group must have at least two members from a different national background. I want you to practice your communication skills!

With groups of three: do make sure that if 2 members have the same native language that they don't exclude the third person!

Deadline for registering teams with me (by email): April 6.

Project presentations: July 1.

For the presentations, you will need the documentation and the implementation of the system. The system should be demonstrated in a live demo.

More tasks will be added for the project as we progress with the class contents. For now, please do these tasks with your team mate(s):

1. Find a subject for your database, i.e. what data would you like to store in it. Choose something that you like or that interests you. Be creative, have fun.
Consider that you will eventually need to input data into your database, so pick only a subject for which you will be able to collect data.
2. Create an ER model and a UML model for your application.
The total sum of the number of entities plus the number of relationships in your model should be not less than 6 (in a group of three people: not less than 9). Keep in mind that eventually you will have to enter data for all entities to test the system.
3. Define three interesting queries (groups of 3: five queries!) that you would like to ask your database once it is filled with data and ready to be used.
4. Transfer your model to a relational database schema and make decisions about normalization.

5. Define suitable integrity constraints.
6. Make assumptions about the quantity of data and the types of data access and suggest a physical design for your system. Explain your design decisions.
7. Document your ideas so that you and your teammate have something to work with in the next steps.