Part 1, 10 / 100 Points

The goal of the project is to write a software application which will use a database in a non-trivial way. Form a group of 1 — 3 persons, decide on a project, and write-up a project proposal. The project proposal must contain:

1. The team member's names.
2. The team name. (This is the most important part!)
3. Describe the software project in roughly three paragraphs. Address the following questions:
   (a) What is the purpose of the software?
   (b) Why is the software interesting?
   (c) What existing product is similar to the project you are proposing?
4. The schemas for the database the software will be using. Anywhere between 7 — 12 is reasonable for a small project, but the number can fall outside of this range.

Part 2, 90 / 100 Points

The project you turn in on the due date should bare some resemblance to what you proposed in part 1. The schemas of the database you turn in do not have to match what was proposed, but the general purpose of the software should be the same. You will need to present your work to the class as a group.

Presentations will be graded on the following criteria

Content	Clearly explain the purpose and design of the software and database.
Demonstration	Demonstrate the use of the software and important features.
Verification	Show the state of the database before and after the software has created, updated, and possibly deleted records.

Your project must use a relational database in a non-trivial way, and must have a graphical user interface which allows the creation, updating, and possibly the deletion of records from the database. Submitted projects will be graded on the following

Database Documentation	The database design must be recreatable from SQL scripts, and the schemas must be documented and accompanied by ER diagrams.
Software Documentation	The software should be accompanied by a document telling me how to use it, and how to verify its correct operation.
Software Correctness	While the software does not have to be "bug free", I should not encounter bugs while performing the documented operations, or variations thereof.

Here are some ideas to get you started:

Library Control

Some tables might include:

1. A table of all books the library contains, (and possibly books contained by partner libraries which are eligible for interlibrary-loan.)
2. A table of library members and their contact information.

3. The history of every member: when they checked out a book, when it was returned, etc.

4. etc.

Remember that a library might also include scientific specimens, musical instruments, DVDs, etc.
The software would need to allow a librarian to add new books, remove lost or damaged books, checkout a book to a member, etc.

School Management

Clone the relevant parts of CUNYFirst. Tables might include: classes, students, instructors, enrollments, etc. The software should allow an administrator to create classes and students, and it should allow students to enroll in classes.

Social Media

Design a clone of your favorite social media platform, or make a new one of your very own.

Scientific Database

Design a database to store scientific data of some sort. For example, a citizen science project which logs bird sightings around the world, similar to `http://ebird.org`.