COMS W4111.001-Introduction to Databases
Spring 2021

Project Part 1

(worth 20% of overall Project grade)

FAQs

Summary of Deadlines

- By **Jan 20:** Find a teammate or email Debbie if you need help finding one.
- During the week of **Mon Jan 25 through Fri Jan 29:** Meet with an IA to discuss your application and design (details below). **This is a required meeting.**
- By **Feb 10:** Submit Part 1 of your project electronically (details below).

Teams

You will carry out this project in **teams of two**. If you can't find a teammate, please follow these steps:

1. Post a message on Piazza asking for a teammate - **the best way**.
2. Send email to Debbie right away (and definitely **before Jan 20**) asking her to pair you up with another student without a teammate. Debbie will do her best to find you a teammate.

You do not need to notify us of your team composition. Instead, you and your teammate should assign yourselves to an available "Project Group" on CourseWorks. For this, go to the "People" section of CourseWorks for our course. Please join one of the **already-created "Project Group n" groups**. Please do **not** create your own group. (We have already created sufficiently many such groups to accommodate all students in the course.)

Important notes:

- If you decide to drop the class, or are even remotely considering doing so, please be considerate and **notify your teammate immediately**.
- On a related note, do not wait until the day before the deadline to start working on the project, just to realize then that your teammate has dropped the class. It is your responsibility to start working on the project and spot any problems with your teammate early on.

You will do the **project** in teams of two students. Both students in a team will receive the same grade. Team partners are expected to fully collaborate with each other on solving the project. However, communication about project details with somebody other than your partner is not permitted, and is considered cheating. Your project submissions--including all programming code--should be your own team's work. Anyone found submitting another person's work will be dealt with under the Computer Science Department's procedures regarding academic honesty (see below). You should clearly document in your submission any external code libraries not developed by you that you use in your projects. If in doubt about what libraries are acceptable, please contact the class staff to clarify. You are **not allowed to use any code at all from other students**, even for parts of the projects that you might consider as "not essential." Again, your project submissions should be fully your own team's work, with the exception of documented external libraries that you clearly disclose in your submission.

If in doubt about what kinds of collaborations or consultations are allowed, please check with the instructor immediately. Please see the policies and procedures regarding academic honesty (http://www.cs.columbia.edu/education/honesty) for further details.

Overview of the Project

You will build a substantial real-world database application of your choice. This project is split into three parts:

- Part 1: You will come up with an application of interest to you and you will design the associated database.
- Part 2: You will implement your database design on PostgreSQL, including example data.
- Part 3: You will have two options, namely, **(3.a) Web Front-End Option** or **(3.b) Expanded-Design Option**:
  - **Option 3.a:** If you follow the **Web Front-End Option** (which is the most fun option!) for Part 3 of the project, you will write an application in Python that manipulates the database through updates and queries, through a simple web front-end.
  - **Option 3.b:** If, instead, you decide to follow the **Expanded-Design Option** for Part 3 of the project, you will substantially expand your database design from Part 1 (see guidelines below), without having to develop a web front-end and hence no programming is needed. You will also have to incorporate this expanded design into your database of Part 2.
- Part 4: You will extend the schema of Part 1 with object-relational features.

Parts 2, 3 and 4 will be based on your description and design of Part 1.

Overview of Part 1

Part 1 of this project consists of multiple steps. At a high level, you will:

- Find a teammate.
- Select an application and write a short proposal.
- Meet with a course staff member to get feedback on your proposal.
- Revise your proposal and then create the full entity-relationship diagram and corresponding SQL schema for the database.

Pick an application that you will enjoy, since you will be working on it for a substantial part of the semester! A suggestion is that you build a database about something that you are interested in --a hobby, a favorite website, material from another course, a research project, etc. It's especially nice if you pick an application where you can populate your database using real, as opposed to fabricated, data. If you have an application where you can get a large amount of real data to populate your database, all the better, but it's not necessary.

If you're having trouble thinking of an application, take a look at almost any popular website. These sites tend to have similar themes and many could be reduced to an appropriate scope for the project. For example, social networks (e.g., Instagram, Reddit, Twitter), shopping sites (e.g., Etsy, Amazon), or content sites (e.g., The New York Times) can all be appropriate models. In particular, and as an example, shopping sites all have a similar theme: products, customers, orders, shopping baskets, reviews, etc., and typically could be reduced to make for an interesting and appropriately sized application. Try to make your application interesting, including a variety of different kinds of attribute domains (e.g., strings, integers, etc.) and relationships (i.e., with different key and participation constraints).

What you need to do for Part 1

1. Find a **teammate.**
2. Decide on an application for your project and **write a document** with the following items to bring to your meeting with an IA during the week of Jan 25 through Jan 29:
   - **A relatively informal, one-paragraph description of the application, not to exceed 20 lines or so**, highlighting interesting and challenging parts. You can ignore all "security"-related issues (e.g., user authentication, encryption) in your application. The more concrete your written description, the more efficient and useful the meeting with the class staff will be. This paragraph should include:
     a. A high-level description of the general domain of the application.
     b. An idea of what entities and relationship sets you will have, including attributes and constraints. You don't need to have your design completely finalized, though, but of course it will help if you bring to the meeting at least a **preliminary entity/relationship diagram**, so we encourage you to do so. You should pick an application with a schema that is relatively substantial, but not too large. As general guidelines, your E/R design should be expected to have around 5 to 10 entity sets and around 5 to 10 relationship sets. You will get a sense if your design is too simple or too complex. Please talk with an IA during office hours if you are in doubt about this.
     c. A specification of whether you will follow the Web Front-End Option or the Expanded-Design Option for Part 3.
   - **A brief "data plan,"** providing just two or three high-level sentences explaining what data you will use to populate your database later on.
   - **(Web Front-End Option) A description of your user interaction plans**: If you will follow the Web Front-End Option for Part 3, you should provide specific details as to how users will interact with your application. For example, if your application is somehow inspired by the Internet Movie Database, your description should describe the general "entities" that are involved, plus explain that your application might ask users for a movie title and return as a result the actors in the movie; you might also let users store in the database the fact that they liked certain movies and disliked others; finally, given a userid, your application might give recommendations on the movies that the user might like, given the user's previously recorded preferences, according to some simple "recommendation" algorithm.
   - **(Expanded-Design Option) A description of your design expansion plans**: If you will follow the Expanded-Design Option for Part 3, you should write a relatively informal, one-paragraph description of how you will expand your design in Part 3, not to exceed 20 lines or so. The expansion of your project in Part 3 should augment your project --in terms of the number of entity sets, relationship sets, and overall "complexity" of the design-- roughly by 50%. This expansion should be substantial: rather than just adding a few entity sets and relationship sets that are overly similar to those in Part 1 of the project, you are expected to add a truly novel and significant component to your database (following the above "50% increase in complexity" guidelines). For example, if your Part 1 database follows some variant of the Amazon shopping site, a substantial expansion for Part 3 could be the addition of a sophisticated "subsystem" for product reviews and ratings, as well as for allowing users to vote on the usefulness of the reviews from other customers, etc.
   - **A short description of your contingency plan**. Since students do occasionally drop classes, and to prevent last-minute surprises, we suggest that you bring to the meeting a written "contingency plan" for the unfortunate case in which one of the teammates drops the class later in the semester. This contingency plan should indicate how you will "downgrade" the project to a simpler one in such a case **--including in Part 3, whether you follow the Web Front-End Option or the Expanded-Design Option--**, so that it is appropriate for a single person to complete. As general guidelines, your E/R design for a one-person project should have around 3 to 7 entity sets and around 3 to 7 relationship sets. If your teammate drops the class, rather than finding a new teammate to complete the project, which is problematic for a number of reasons, you will complete the "downgraded" version of your original project. Including such a contingency plan is optional, but if you choose not to bring it to your meeting in written form, for approval, and your teammate drops the class later, you will have to complete the original project as planned, and no exceptions will be made at that point.
3. **Meet with an IA during the week of Jan 25 through Jan 29** to discuss your design and plans, and make sure that they are appropriate (i.e., challenging enough, but not unrealistically so). **This meeting is required** and should last up to **10 minutes**. Your grade for Part 1 will be decreased substantially if you don't meet with any of the class staff. We will have **expanded office hours** during that week and you must sign up for a meeting slot ahead of time (details of the sign-up process will be announced soon. **Both team members must attend** the meeting, at the same time. In advance of this meeting, email a **pdf copy of the document** described in the previous point (i.e., in item (2)) to the IA.
4. After an IA has OKed your general application, your data plan, your plans for Part 3, and your contingency plan, submit the following three items **electronically on CourseWorks** just once per team (not once per teammate) by Feb 10:
   - A **revised version of the document** that you brought to your meeting with the IA (including the description of your application, your data plan, your plans for Part 3, and optionally your contingency plan); this revised version should incorporate all of the feedback that you received during the meeting.
   - The **E/R diagram** of your database, following the syntax that we saw in class (consistent with the textbook). You should specify as many of the real-world constraints for your application as possible. Write in plain English at the bottom of the diagram any constraints that cannot be captured with the E/R syntax that we use in our course.
     Note for Expanded-Design Option: in Part 1, you do not need to write or submit an E/R diagram for your Part 3 expanded design; instead, you should just submit the one-paragraph description of your expansion plans for Part 3, which you should have brought to the meeting with an IA and **gotten approved during the meeting**. You will do the E/R diagram, etc. for Part 3 later in the semester.
   - The **SQL schema** of your database, which you should derive by mapping your E/R diagram into SQL using the method that we will cover in class. You should capture as many of the E/R constraints (e.g., key and participation constraints) as possible. Briefly discuss in plain English any constraints that you cannot yet reflect in your SQL schema.

   Important submission instructions: To submit your project, you need to be in the Class view (**not** the Group view) and then upload your file or files to the "Part 1" assignment under Assignments. Please do not upload .zip, .tar, etc. files, but rather you can submit multiple (uncompressed) files if necessary. (CourseWorks makes the handling of .zip, .tar, etc. files for grading more complex for the IAs.) You should submit your project exactly once per team, rather than once per student.

Note 1: Try to meet with an IA **early in the week** of Jan 25. If you meet with us early in the week, you will still have the chance to attend any of our expanded office hours later that week to discuss further and get answers to any questions that you might have.

Note 2: If you observe religious holidays that overlap with this part of the project, please email the instructor to arrange for alternative deadlines.

Grading for Part 1

Your grade for Part 1 will be split as follows:

1. **Meeting with class staff: 5 points.** If you come to the meeting prepared with a hard copy of your document as specified above, you can expect to get all points, even if you are asked to make changes or revisions to your proposal.
2. **Quality of revised document for your application: 5 points.** We will evaluate the overall quality of your final description of your application, especially in terms of how thoroughly you incorporated any revisions suggested during your meeting with the class staff.
3. **Quality of E/R diagram: 5 points.** We will evaluate how well your E/R diagram models your proposed application, including how well you modeled any relevant real-world constraints.
4. **Quality of SQL schema: 5 points.** We will evaluate how well you mapped your E/R diagram, including constraints, into a SQL schema using the technique that we covered in class.

Frequently-Asked Questions

- **Q: I have a really cool idea for the Project, but in order to implement it I would have to work alone. Can I?**
  A: Please modify your project idea so that it becomes appropriate for a two-person team. Being able to work with others is a necessary skill to develop for the real world.
- **Q: Can my team have 3 (or 4, or 12) students for the Project?**
  A: No, your team has to have exactly two students, so that we can grade all projects as uniformly and fairly as possible.
- **Q: Why use PostgreSQL for Parts 2 and 3 of the Project? Can I use my favorite DBMS instead?**
  A: As much as we would like to be more flexible, we just don't have the staff to handle several diverse systems and platforms. Unfortunately, you cannot use any other DBMS.
- **Q: Can I use Ada (or some other language other than Python) for the Part 3 Web Front-End Option of the Project?**
  A: Please see the answer to the previous question. Also, Python is much easier to work with than other languages such as Java for our project. Furthermore, Python is a great, easy-to-learn, widely used language. If you are fluent in Java, you will be able to easily learn the (not-so-deep) level of Python needed for our project.