COMS 4111 - Introduction to Databases, Fall 11
Project 1, Part 1
(worth 25% of overall Project 1 grade)

FAQs

Summary of Deadlines

- **By Wednesday Sep 21, 9:00pm:** Find a team-mate or email Kritika (kk2759@columbia.edu) if you need help finding one.
- **During the week of Friday Sep 23 through Thursday Sep 29 :** Meet with a TA to discuss your application and design (details below). **This is a required meeting.**
- **By Wednesday October 5 at the begining of class, 1:10pm for Section 1, 5:40pm for Section 2:** Submit a hard-copy with your final project description (details below).

  Important note: Please check the Project Lateness Policy carefully. For Part 1 of Project 1, you can **only** use project grace days for the final hard-copy submission (i.e., you **cannot** use grace days to delay your meeting with a TA or the instructor).

Teams

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

1. Post a message in the class discussion board on CourseWorks asking for a team mate - **the best way**.
2. Send email to Kritika (kk2759@columbia.edu) right away (and definitely **before Wednesday Sep 21, 9:00pm**) asking her to pair you up with another student without a team-mate. Kritika will do her best to find you a team-mate.

You do not need to notify us of your team composition. Instead, when you submit the first part of your project you should indicate in your submission your team composition.

Important notes:

- If you decide to drop the class, or are even remotely considering doing so, please be considerate and **notify your team-mate 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 team-mate has dropped the class or moved to another planet. It is your responsibility to start working on the project and spot any problems with your team-mate early on.
- You might be able to do the project by yourself if you have a compelling reason, but we strongly discourage this. (See below for more details.) This is a large class, and we are relatively understaffed, so we simply cannot handle a large number of individual projects. If you want to do the project by yourself, you should get **explicit permission** by email from Aarthi: Please send her email giving a compelling reason why you want to work by yourself.
  - Not knowing other people in the class is **not** a compelling reason, and you should use the discussion board or ask Aarthi to help you find a team-mate. (See above.)
- Please check the policies and procedures regarding academic honesty for important information of what kinds of collaboration are allowed for projects.

Computer accounts

You will need a **CS account** to do the final parts of Project 1 (not for Part 1, though) but only if you decide to do the project on Unix. (You will not need a CS account if you do your project fully on your Windows laptop.) Consequently, if you use Unix, then at least one of the team members should have a CS account. If neither of you have a CS account, please open one from https://www.cs.columbia.edu/~crf/accounts/cs.html. Choose the appropriate "student" category as the "account type." There is a $50 charge to open a CS account. Please refer to CRF's homepage for details on infrastructure and policies of the CS department.

Overview of Project 1

In Project 1, you will build a substantial real-world database application of your choice. You will do the conceptual design of the database on paper **in Part 1,** and then you will create an actual database on Oracle **in Part 2**. **Part 3** requires some explanation, because you will have two options on how to complete it: **(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 PHP or Java: your choice) 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 with no programming needed. You will also have to incorporate this expanded design into your Oracle database of Part 2.

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

Pick an application that you will enjoy working with, since you will be stuck with 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 web site, 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. As the project progresses, you'll end up creating a database of at least dozens of entities/relationships. 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.

We have instituted a project contest, and that the best projects (as decided by the class staff) will get a **10% boost** in the Project 1 grade.

Overview of Part 1 of Project 1

For **Part 1** of this project, which you should submit **in hard copy**, you should include **3 items**:

1. **(Required) Detailed description of your application:** You will describe the general "domain" of your application, construct an Entity-Relationship diagram for the database, and map it to a relational schema using the mapping technique that we will cover in class.

   Try to pick an application with a schema that is relatively substantial, but not too large. As general guidelines:

   - If you are doing the project in a team of two (which should be the general case), your E/R design should have around seven entity sets and a similar number of relationship sets. This is a ballpark figure only: something in the 5-to-10 range should be fine. You will get a sense if your design is too simple or too complex. Please talk with a TA **during office hours** if you are in doubt about this.
   - If you are doing the project by yourself (you need explicit permission; see above), your E/R design should have around five entity sets and a similar number of relationship sets. This is a ballpark figure only: something in the 3-to-7 range should be fine. You will get a sense if your design is too simple or too complex. Please talk with a TA **during office hours** if you are in doubt about this.

   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). It is important that you include as many relevant constraints for your application from the real world as possible in your E/R diagram.

2. **(Required) Your choice for Part 3 of the project:** As part of your Part 1 submission, you need to specify whether you will follow the **Web Front-End Option** (Option 3.a) or the **Expanded-Design Option** (Option 3.b) for Part 3 of the project (see above).

   - If you will follow the Web Front-End Option, then you should simply say so with your Part 1 submission. In this case, you do not need to include any additional information regarding Part 3 with your Part 1 submission.
   - If you will follow the Expanded-Design Option **or if you are not sure at this point**, then you should state this in your Part 1 submission, as well as **specify how you will expand your design in Part 3**. 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 an additional 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. (See below for more guidelines.)

   If you state in Part 1 that you will follow the Expanded-Design Option, it is OK for you to change your mind and decide to follow the Web Front-End Option later, when we announce Part 3 of the project; however, if you do **not** submit a plan for the Expanded-Design Option when you submit Part 1 of the project, then you will have no choice and you will have to follow the Web Front-End Option; we will make **no exceptions** to this rule.

3. **(Optional but strongly recommended) Contingency plan for two-person teams:** Since students do occasionally drop classes, and to prevent last-minute surprises, we suggest that you also include in your submission a **"contingency plan"** for the unfortunate case in which one of the team-mates 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, if you follow the Expanded-Design Option--**, so that it is appropriate for a single person to complete. (See guidelines for single-person projects above.) If your team-mate drops the class, rather than finding a new team-mate 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 submit it when you submit Part 1 and your team-mate drops the class later, you will have to complete the original project as planned, and **no exceptions** will be made at that point.

What you need to do for Part 1

1. Find a **team-mate.** There's no need to notify us of this; you will simply indicate who your team-mate is when you submit Part 1.
2. If you will do the project on **Unix** machines, get a **CS account** if neither you nor your team-mate has one.
3. Decide on an application for your project and:

    - **Write a relatively informal, one-paragraph description of the application, not to exceed 20 lines or so**, highlighting interesting and challenging parts. If you're having trouble thinking of an application, take a look at any web shopping site (e.g., Amazon). They all have a similar theme: products, customers, orders, shopping baskets, etc., and typically make for an interesting and appropriately sized application. (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 (see below). This paragraph should include:
        a. A high-level description of the general domain of the application. If you will follow the Web Front-End Option for Part 3, you should also provide specific details as to how users will interact with it. For example, if your application is somehow inspired in 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.
        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.
        c. An idea of **what data you will use** to populate your database later on.

    - If you will follow the **Expanded-Design Option** for Part 3 of the project, **write a relatively informal, one-paragraph description of how you will expand your design in Part 3, not to exceed 20 lines or so**. As discussed above, 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%. As discussed above, 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 web 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. If you will follow the **Web Front-End Option** in Part 3, then you do not need to write anything for Part 3 at this point.

    - **Write a short description of your contingency plan** (see above).

4. **Meet with a TA during the week of Friday Sep 23 through Thursday Sep 29** to discuss your design and make sure that it is appropriate (i.e., challenging enough, but not unrealistically so). **This meeting is required** and should last **about five 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. Please show up early in the week to minimize delays. We will **not** be taking appointments, so please show up directly during office hours. **Both team members should attend the meeting, at the same time. For this meeting, bring the written materials described in the previous point (i.e., in item (3)).**
5. After a TA has OKed your general application, your plans for Part 3, and your contingency plan, modify your description based on the feedback that you were given, and write an **E/R diagram** (following the syntax that we saw in class) of your database, specifying as many of the real-world constraints for your application as possible.

    Important note if you follow the Expanded-Design Option for Part 3: 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 a TA or the instructor and **gotten approved during the meeting**. You will do the E/R diagram, etc. for Part 3, later in the semester.

6. Using the method that we will have covered in class, map your E/R diagram into a **relational schema in SQL**, capturing as many of the E/R constraints (e.g., key and participation constraints) as possible.
7. Submit a **hard copy** of **(a)** your one-paragraph description of the application (which you should have revised based on the feedback from your meeting with one of the class staff), **(b)** your E/R diagram, **(c)** your resulting SQL schema, **(d)** your one-paragraph description of your expansion plans for Part 3, if you are following the Expanded-Design Option (which you should have revised based on the feedback from your meeting with one of the class staff), and **(e)** your contingency plan (again, revised as appropriate based on your meeting with one of the class staff). You should submit all these materials at the beginning of class **(1:10pm for Section 1, 5:40pm for Section 2) on Wednesday October 5**.
8. **Keep a copy** of all these materials for yourselves, since you will need them for Parts 2 and 3 of the project.

Important note 1: Try to **meet with a TA early in the week of Friday Sep 23 through Thursday Sep 29**. If you wait until the last day of this period to do so, you will only have a little more than a few days to complete Part 1 of the project.

If you meet with us early, you will still have the chance to attend any of our expanded office hours later to discuss further and get answers to any questions that you might have.

Important note 2: If you observe religious holidays that overlap with this part of the project, please email the instructor (biliris@cs.columbia.edu) to arrange for alternative deadlines.

Grading for Part 1

Your grade for Part 1 of Project 1 will be split as follows:

1. **Meeting with class staff: 7 points.**
   If you come to the meeting prepared with your written description as specified in items (3) and (4) above, you can expect to get all points, even if you are asked to make changes or revisions to your proposal.
2. **Quality of final one-paragraph description of your application: 6 points.**
   We will evaluate the overall quality of your final hard-copy one-paragraph 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: 6 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 your SQL schema: 6 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 Project 1, but in order to implement it I would have to work alone. Can I?**
  A: Unfortunately this is not a "compelling" enough reason for a one-person team. Please modify your project idea so that it becomes appropriate for a two-person team.
- **Q: Can my team have 3 (or 4, or 12) students for Project 1?**
  A: No, your team has to have exactly two students (unless you get explicit permission from Aarthi for a one-person team; see above).
- **Q: Why use Oracle for Parts 2 and 3 of Project 1? 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 PHP or Java) for the Part 3 Web Front-End Option of Project 1?**
  A: Please see the answer to the previous question.
