Each of you has had a topic assigned for this project based on your submission for assignment 2. You can locate your topic in the grading comments for assignment 2. In the next few assignments you will build, populate and query your database. Please do NOT work ahead. I will grade each assignment and make suggestions for corrections before you begin the next assignment. If you work ahead, you might have to delete your work and start over. I will try to grade them as quickly as I can. The sooner you turn in the assignment; the sooner you can start on the next part. If everyone turns in the assignment at the last minute, you will have a very limited time to do the next part. Keep this in mind when scheduling your time for this course.

For the first part of the project, you are going to do two important tasks. First, you have to develop the requirements upon which the system will be based. In order to do this, you will create three uses (separate users or separate uses if yours is not a corporate type database) for your database. For each use/user create requirements that must be met. These requirements are not just a list but a detailed explanation of each requirement and why it is important. Requirements may be data needed for the user to do his/her job or formatting or domain requirements that must be enforced. This is not a list of what queries the user might want answered. Frequently the requirements for different user groups conflict. Make sure you specify any constraints that must be implemented to meet the different user's requirements.

Once you have created the three sets of requirements and constraints, combine them into one set of requirements. Make sure you note what compromises you had to make to come up with a single set of requirements.

Finally, create three or more relations needed to implement your requirements. For each relation create a table using the following format. Do not create a different format because it is like the one you use at work or you think it is better. A consistent format in everyone's project speeds the grading process. Please remember to list the primary key(s), foreign key(s), and indices below the table. If there is not one, type the label and leave the information blank. A foreign key must list the table it references and if the attribute names are different in the two tables then it needs to list the attribute as well. For example, facultyID references FID in FACULTY. facultyID is the attribute name in the relation you are describing and FID is the attribute name and a primary key in the relation FACULTY that you are linking to. Do not index your primary key. Only index attributes that might be used frequently in queries but are not primary keys in the relation.

RELATION NAME				
Attribute Name	Data Type	Size	Constraints/Domains	Cascading Problems

Primary Key(s):

Foreign Key(s)

Indices:

The more thought you put into this assignment, the easier the others will be. Please submit as a Word document so I can use tracking changes to make comments. Remember to use page headers and footers as you have on your other assignments. Make sure that where you have been putting the assignment number, you now put the project number and your topic. For example,

Project 1 - University

in the center of the page header.

For those of you who need more direction on system requirements, think about our university system. First you think about the functions the system will perform and any forms that might already be used. For example, a Change of Enrollment form is used to register for courses at the university. It also allows for instructor overrides. On this form at Stevens, the course is two alphabetic characters followed by three digits. This format for the data is a system requirement. All of these functions and domain restrictions must be addressed in the requirements section of your paper.

Three user groups could be students, faculty, and administration. Each group has a different reason for using the system and frequently need different attributes to reach their goals. Students need information about courses and grades. Faculty need information about courses, departments and advisees. The administration needs information about faculty and facility usage. Students want to see their grades as letters but also would like to see a GPA. As a faculty member, I would like to see grades as numbers. How do we decide how to store them? In other words, what is the restrictive domain for the grade field. State each user's need and then in the consolidation phase, state what you will use as the final format and domain and how you will handle the needs of the other users. We do not want to store both an A and the number 4. Which is preferable and how do you handle the conversion? There is no right answer but a compromise must be made to make the requirement for the database work for all users.

Sometimes two user groups require different domains. Address the issue before creating the database. Remember it will make your future work easier if you pick user groups that will need different queries.

These are by no means complete requirements. They are just a little insight into what you must do for each of your distinct user groups.

By the time you finish the entire project you will have formalized your requirements, built the relations, inserted data and queried the database from the prospective of the different user groups.