CS 3743
Introduction to Database Systems
Project

In this project, you will design and implement a database application in stages. You will identify an application in the real world, design a conceptual database scheme, specify it to a commercial database management system, insert data, write interactive queries and update the database using SQL, create constraints and triggers, write application programs using JAVA/JDBC and PL/SQL and create a web interface for the database.

You can work on teams of two to three. Form your teams as soon as possible. All the team members will get the same grade for the project. The project is divided into four stages. Each stage has a separate due date. No late projects are accepted. You should complete each stage on time.

The project consists of the following stages

3. Create, populate and query your database.
4. Database programming using PL/SQL, Pro*C/C++ or Java/JDBC

1 Conceptual Design: Due Monday, February 18

Identify a real world application. Your application should support multiple types of users and multiple activities per user. Following is a list of application domains that can give you an idea.

- **Movies:** Maintain information on actors, directors, genres, playing times, movie theaters. You might have queries such as “Find movies directed by X” or “Find movies in which actor Y” appears. Check out the webpage http://www.imdb.com to get an idea of what can be done.

- **Apartments:** Maintain information on apartments and their properties. You can provide an interface for offering apartments for rent or finding apartments. Check out the webpage http://www.rent.com to get an idea of what can be done.

Submit the following for the first stage.

1. **Cover Page:** Find a title for your project. Write the title of team and the team members on the cover page.

2. **Description of the application:** Describe the types of users and functions supported for each user. For example, consider a university domain. Types of users include students, faculty
and staff. Students can register for classes, view course information and their grades. Faculty has access to course list and assign grades to students. Staff at fee office can access financial information about students. Your application must be broad enough to support multiple types of users and wide range of activities.


4. Potential data sources to populate your database: Investigate how you can get data to populate your database. Using real world datasets are preferable since they correspond to real world database applications. SQL has a utility to load data from delimited text files to your database. Your data sources could be text files with attributes separated by some marker such as space or comma.

## 2 E-R Diagram, Mapping to relational model and normalization: 
Due Wednesday, March 20

In this stage, you will create an E-R diagram for your application, map your E-R diagram onto relational model and normalize the relations.

- **E/R Diagram of your database:** Formulate a requirement list for your application and draw the E/R diagram for the application. If there are rules that can not be specified using E/R diagram, list them on a separate piece of paper.

- **E/R to Relational Mapping:** Use the mapping algorithm to translate your diagram into relational schema.

- **Functional Dependencies:** For each relation specify a set of functional dependencies based on semantics of the data. You should use knowledge of application domain to identify the functional dependencies.

- **Normalize Tables:** For each relation identify its normal form. For relations not in BCNF, perform the normalization. Your normalized tables should satisfy lossless join and dependency preserving properties. If you fail to normalize to BCNF with lossless join and dependency preserving properties, then normalize to 3NF.

Submit the following for the second stage.

1. **E/R or EER Diagram**

2. **Final relational schema of your database.** For each relation, list the set of functional dependencies that the relation satisfies. List all the candidate keys and the primary key. Indicate the normal form of each relation and the foreign keys.

3. List any constraints that your application requires, but can not be specified directly in the relational schema.