Phase I

Advising can be a complex and time-consuming process, especially in light of the ever-changing degree requirements, general education requirements, and other disciplinary, college, and campus requirements.

In order to help advisors and their students, a multi-campus university has made a request for the implementation of an ADVISING information system. The purpose of this system is three-fold. First, to provide access to timely and easy to use advising information for students. Second, to provide the academic advisors with the ability to provide accurate advise to their students. The third purpose of this system is to provide the university administrators and academic units an easy to use tool for entering and maintaining degree program requirements. It is important (in fact crucial) that your system captures the degree requirements entirely as data relationships as opposed to program logic.

The new system must provide facilities to access, insert, modify and remove information about the following:

- Students, their major, courses taken, or advised to take in the future.
- Academic departments, degree programs and their requirements.
- Advisors and their advisee comments.

**Goal:**

Our goal is to design the database which can support a 3 tier scalable web based ADVISING
information system.

**ERD:**

As the basis for our design we will use the ERD which we developed and refined during class. (A large scale version of this ERD can be found on our class web site)

*Note: Revisions and Corrections to the UNIFIED ERD may appear on our class web site. It's your responsibility to check the site regularly.*

**Functional Requirements:**

As mentioned earlier the system must support three different groups of users (Students, Advisors, and University Administrators). The following functional decomposition diagram will provide a high level view of the functionality envisioned for this system. In a properly designed system, database queries are used to satisfy all the data-access requirements of the system. Later, the data retrieved from the system will be manipulated and rendered by the user interface.

**Part - 1 Update Operations: (6 Operations)**

- Using the ERD and entity attribute table (available on our web site) provide 6 examples (2 Insert, 2 Delete and 2 Modify) of relational algebra update operations. Identify the 5 relational constraints discussed in class as they relate to these operations. For example when you provide an example of the Insert operation, indicated what constraints may be violated, and what must be done to avoid potential problems.

**Part - 2 Retrieval Operations: (6 Queries)**

Pick one of the 3 sub-systems shown in the above decomposition diagram. Develop at least 3 trivial and 3 non-trivial questions for the sub-system which you have selected. (In my book, a trivial query is defined as a query that uses one or two tables. A non-trivial query is defined as one using three or more tables, with multiple conditions).

- Using relational algebra operations such as Select, Project, Cartesian Product, Union, Intersection, Difference, Division, Left Outer Join, Right Outer Join and
Join, write the above 6 queries in relational algebra operators. Make sure to use at least 6 of the relational algebra operations in your queries.

- Example Queries:
  - Trivial: List of all the Computer Science students who entered the BS or AS degree program after Fall 2006.
  - Non-Trivial: Display the above students’ degree audit.

**What to hand in:**
- Cover page with paper title, your name, course # and name, assignment #, date.
- ER diagram for the above application. (Note the modifications made to the ERD discussed in class.)
- Entity-attribute-table for all entities as well as associative entities. (Note the modifications to attributes, keys, foreign keys, multivalued attributes, composite attributes and derived attributes.)
- Your Queries (in English as well as Relational Algebra)

**Optional:**
- Using a drawing package develop a set of potential user interfaces for this database. (Input and output screens.)