Project: Grad school Database

1 Purpose of the project

You are to analyze the requirements for, design, implement, demonstrate and document a database system that could automate the administrative functions of a graduate department, which actually is very close to our own CS department. The system should keep track of graduate students, professors, courses, exams, and it should generate the necessary reports.

2 Requirements

The major user of this system is the Graduate Office Coordinator (“GOC” from now on), who will be doing most of the data entry. A description of the major functions and data items follows.

Data: The system should include information about the graduate students, the professors, the courses, the thesis proposals and defenses.

- **Student data**: For every student, we want to record the full name, the starting semester, the social security number (notice that it may change - foreign students originally get a temporary ssn), the advisor/co-advisor history, the courses and grades the student has taken.
  - Students have to choose an advisor (and often, a co-advisor); they may change advisor/co-advisor as often as they want - we record the date of each change, and the name(s) of the new advisor(s).
  - Students have to take courses: 5 of them should be “star” courses (designated so), one from each of the five areas each (Algorithms and theory (TH), AI, Computer Systems (CS), Software Systems (SS), Programming Languages (PL)). They also have to take at least 3 more, electives.
  - Students also TA 2 courses (or more, if they want to); we store the semester and course number.
  - Students may choose to go on a leave of absence (LOA), for a year or more.
  - Students also have to pass 3 exams (Speaking, Writing and Programming skills) - we record the date they passed it.
  - Students also have to do the thesis proposal, and we record the date, title, and the faculty members of the committee (at least 3).
Finally, they do the thesis defense, and we record the date they pass it the title (may be slightly different from the thesis proposal), and the members of the committee (at least 3 professors from the department, at least one external from another department/school; not necessarily the same as the ones from the thesis proposal)

Once a student graduates, we record the fact, and store his/her forwarding address and new employer, if known.

In the rare case that a student decides to drop out, we record the fact.

For each student, we occasionally want to record some notes (eg., "ACM dissertation award winner")

• **Course data:** For each course we store the course number (unique, at that time frame), the title and the syllabus. New courses may be introduced; old courses may be "retired", and their course number may be re-used.
  
  Each course may be offered in zero or more sections (numbered 'A', 'B', etc), in each semester; there may be 1 or more instructors (co-teachers), even for a single-section course
  
  Some courses are designated as “star” courses - we have to record this fact.
  
  for each course offered, we record the instructor(s).
  
  There are only two semesters, “Fall” and “Spring”, that we care to record.
  
  We also record the TAs for each course - they are usually our CS graduate students, but occasionally, external people may be hired.

• **Professors:** We record the full name of each professor, the teaching history, the advising history. Notice that the name might not be unique ("John Smith"); it might even change, say, due to marriage. New professors are hired, usually on a yearly basis; some professors leave (e.g., they retire).

**Tasks:** The following type of events should be handled by the target system:

1. **Registration:** a new student arrives - we record the name, gender, ssn, and mailing address; also the undergraduate institution, the year of graduation, and, if the student already has a graduate degree, the graduate institution, the name of the degree (eg., "M.Sc.", "MBA"), and the year. We also record the area of interest, one of the five areas: (‘TH’, ‘CS’, ‘SS’, ‘PL’, ‘AI’, or leave it null, if undecided)

2. **Changes:** Rarely, a student may change his/her ssn from a temporary one to the final one. We only record the final one. The student may also change the area of his/her interest.

3. **Advising:** A student picks an advisor (or two), or changes his/her current advisor(s); we record the new advisor(s) and the date of the change.

4. **Grades:** A student takes courses; we record the letter grade ("A+", "A", ..., "C", "C-", "F", or just "P" for Pass) the semester, and the course number

5. **Skills:** Whenever a student passes one of the three skills, we record the skill-id (SPK, WRT, PRG) and the date.

6. **TAing:** Whenever a student TAs a course, we record the course number, section number and the semester (eg., 15-415, A, Fall 2002)

7. **LOA:** Rarely, a student decides to take a leave of absense for a year (renewable for more years, after negotiations) For each year of LOA, we record the fact.
8. Thesis proposal: Whenever a student passes his/her thesis proposal, we record the date, title, and the composition of the faculty committee (3 or more professors from our department).

9. Defense: Similarly for the final defense, with the addition that there is an external member of the committee (a professor in some other university, or some research lab). We also mark this student as 'graduated' and record the forwarding address and new employer, if known.

10. Drop-out: Very rarely, a student decides to drop out of the program. We mark his/her record as 'terminated', and optionally fill in some notes (eg. reasons for termination).

11. Course scheduling: We record which courses will be offered next semester, in how many sections, which professor(s) will be the instructor(s) for each section, and which graduate students will be the TAs (zero or more TAs per course). We can also make changes (eg., a scheduled course might not be offered in the last minute, because of an emergency).

12. Introduction/retirement of courses: Occasionally, new courses are introduced: the G.O.C. assigns a course number to them; we record the title and syllabus; the date the course was introduced, the area of the course (CS, AI, TH, SS, PL), and whether it is a 'star' course. (Recall that students need to take 1 'star' course in each of the five areas mentioned above). Also, occasionally, a course is 'retired': we have to mark it as such, along with the date of 'retirement', and return its course number for re-use. Course numbers should be unique for all the active courses.

13. Faculty changes: We want to record new professors that are hired, along with their starting date; we mark retiring/leaving professors as such (“gone”), along with the leaving date. We may also keep free-text notes for each professor (e.g., ”Turing Award winner”)

14. Ratings: Students can login to our system and give ratings for courses they have taken, in the range 0-10; we record them, along with the student-id. They should NOT be allowed to give ratings for a course that their current advisor taught. Alternatively, the G.O.C. could record these ratings, subject to the same restriction.

15. Semester grand report (“Black Friday”): We want to produce the full record of every active student (“active” = still in our program, neither graduated, nor drop-out). For each student, we want:
   - name, starting semester (eg., “Fall 1999”), and area.
   - list of ‘star’ courses taken so far, with the code of the area ('TH', 'AI' etc), the semester and grade;
   - list of the electives taken so far, with semester and grade.
   - the full advisor(s) history, ending with the current advisor(s).
   - the courses (if any) that he/she has TA-ed so far
   - the skills passed, along with the date
   - the date and committee of the thesis proposal
   - the date and committee of the defense
   - any other free-text notes (Leave of absense info, awards, etc)

The report should be sorted according to years in school (excluding year(s) that a student spent on a leave of absense 'LOA'); within the same year, it should be sorted alphabetically.
16. **Rule check:** To help the graduate office coordinator (GOC), we want to implement a task that will check whether a graduating student has fulfilled all of the following requirements:
   - Star-courses: the student must have taken five ‘star’ courses and passed them (’P’, or ’B-’ and above), one in each of the five areas mentioned above
   - electives: he/she must have passed three more courses
   - skills: he/she must have passed all three skills (Speaking SPK, Writing WRT, Programming PGM)
   - proposal: he/she must have passed the thesis proposal
   - defense: he/she must have passed the defense

17. **Course offerings:** For a given semester, we want to print all the courses that we plan to offer, along with the sections, and their instructor(s).

18. **Professor load:** For a given period of time (say, “1/1/2001-6/15/2001”), and for each professor that was still in our department then, we want to list all the students they were his/her advisees, and all the courses the professor taught. Students that are co-advised, should be marked with an asterisk; ditto for courses that are co-taught.

19. **Average program duration:** We want to find the average time it takes a student to complete the program. Specifically, we want the average program duration, for all the students that started in a given era (say, 8/1/1990 - 8/1/1995), and we want to break down the statistics
   - and, in a different report, by advisor.

(In these reports, ignore students who dropped out, or who have not graduated yet)

20. **Histograms:** For the years of program duration, we also want the histograms (ie., number of students that completed in 1 year, 2 years, 3 years, etc.) Prepare one set of such histograms for any given time period,
   - for each gender (male/female)
   - for each of the five research areas (TH, AI, CS, SS, PL)
   - for each advisor
   - for each undergraduate school (report only the 10 most populous ones)

In your histograms, go up to 10 years of studies, reporting the rest (and the drop-outs) in the 11-th category of ”≥ 10”

21. **Most liked courses:** For a given time period, list the courses in their sorted average score (according to grad student scores).

22. **Most liked teachers:** Repeat for the teachers/professors: find the average score that each professor has collected, from all the courses he/she has taught or co-taught; list the professors in descending score order.

3 **Phases**

The three phases of the project cover the following work-processes from the *Adaptable Methodology for Database Design* by Roussopoulos and Yeh [IEEE Computer, May 1984]:

- **Phase I:** “Environment and Requirement Analysis” and “System Analysis and Specification”
• Phase II: “Conceptual Modeling” and “Task Emulation”
• Phase III: “Implementation” and “Testing”

3.1 Reports:

A TYPED report should be handed in for grading at the end of each phase. The final project report consists of three parts: the Phase I report, the Phase II report, and the Phase III report. Each report will be graded out of 100 points. Numbers in square brackets indicate the points (out of 100) for each part of the reports.

The Phase I report must contain
1. [2pt] a description of the purpose of the project and the purpose of this phase of the project.
2. [13pts] the assumptions that you have made about the enterprise.
3. all the documentation produced in this phase, i.e.
   • [30pts] the top-level information flow diagram, (very important - also, don’t forget the system boundary)
   • [5pts] the list of tasks and subtasks,
   • [15pts] the task forms,
   • [5pts] the list of documents, and
   • [30pts] the document forms (very important)

The Phase II report must contain
1. [1pt] a description of the purpose of this phase of the project.
2. [4pts] the top level information flow diagram from Phase I, as well as a list of any revisions that were made to the specification described in the Phase I report
3. the documentation produced in this phase, i.e.,
   • [20pts] the graphical schema using the E-R model,
   • [5pts] list of the attributes for each entity and relationship,
   • [5pts] explanations of the non-obvious entities and relationships,
   • [20pts] the schema in the relational model, in some appropriate Normal Form: BCNF, or 3NF at worst.
   • [5pts] explanations (e.g., primary keys, additional functional dependencies, explanations why a table is not in BCNF e.t.c.),
   • [10pts] the DDL statements to create the above relational schema,
   • [30pts] the code for each task: Pascal-like pseudo-code and the embedded DML code.

The Phase III report must contain
1. [20pts] a description of the purpose of this phase of the project, the relational schema definition from Phase II, and any revisions made to the specification described in the Phase II report.
2. the documentation produced in this phase, i.e.:
   • [20pts] a source program listing.
   • [10pts] a user’s manual for the system.
   • [40pts] your testing efforts: erroneous cases, that your system can detect and handle reasonably.
   • [10pts] a description of the system’s limitations and the possibilities for improvements.
3.2 Grading - due dates

The due dates are as announced in the schedule. The weights are as follows:

- 30% for the report of Phase I
- 30% for the report of Phase II
- 10% for the report of Phase III
- 30% for the Demo.

4 Clarifications in the R+Y methodology

This discussion clarifies or simplifies some points of the methodology by [Roussopoulos and Yeh, IEEE Computer, May 1984]. In your project you should use the above methodology, with the following exceptions/modifications:

1. Ignore queries, symbolized as ‘circles’ in the paper. Replace them with tasks (‘ovals’)

2. **VERY IMPORTANT:** Do **not** show consecutive tasks, **nor** consecutive documents in your top level diagram. That is, a task should be linked to documents only; a document should be linked to tasks only.

3. You may use the list of construct in your Document Forms. For example, a student record could be:

   Student record
   SSN
   Name
   Courses: list_of
     course_name
     grade
     ...