CS348: Introduction to Database Systems

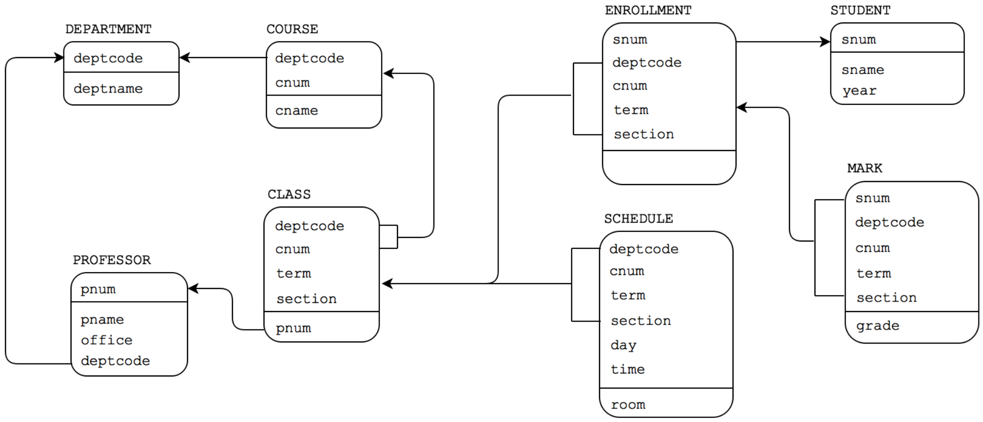
(Fall 2018)

Assignment 2 (due by 12noon on Friday, October 12th)

Overview: For this assignment, you must use your Unix accounts and DB2 to compose and evaluate a number of SQL queries over the same enrollment database as was used in your first assignment. The visualization of the database schema for this database is reproduced below. All submissions must also use the SQL DDL code for this database given in the contents of a file downloadable from the course web page.

As with the first assignment, you are given a requirement for each query in English, and your task is to write source code in the SQL query language that implements the requirement.

Assignment submission: Instructions for submitting this assignment will be given on the course web page.



Note that some the requirements stipulate conditions on what features of SQL may be used in your source code, e.g., that `group by` clauses and aggregate functions may not be used. Part of the grading for your answers in these cases relate to these conditions. The rest of grading will be based on two additional criteria: (1) correctness, the query implements the requirement, and (2) readability. Consequently, the efficiency of your source code, e.g., as determined by DB2, will not be a factor in any grading.

Queries that may *not* use aggregation in SQL

1. The student number and name of students in at least their second year who have obtained a grade higher than 90 in at least two courses their previous year.

2. The number and name of professors who are not in the pure math (PM) department, and who are teaching CS245 for the first time.

3. The number, name and year of each student who has obtained a grade in CS240 that is within 3 marks of the highest ever grade recorded for that course.

4. For each pair of classes for the same course that were taught in the same term, and that where also taught by different professors: the minimum grades obtained and the maximum grades obtained. In addition to these four values, each result should include the number and name of each professor, as well as the identifying attributes for each class.

5. The number and name of each professor who has taught a CS 245 class in which exactly three students received the highest grade and in which every other student received a grade within 20 marks of the highest grade.

Queries that *may* use aggregation in SQL

6. The number and name of each professor who has taught a CS 245 class in which exactly three students received the highest grade and in which every other student received a grade within 20 marks of the highest grade.

7. A count of the number of different students in each term for any course that has never been taught by either a computer science (CS) or applied math (AM) professor. Each result should identify the course, the term and said count, and should be sorted in a descending order by the said count.

8. For each term, the percentage of enrollments in classes for any course that has never been taught by either a computer science (CS) or combinatorics and optimization (CO) professor. The result should be in chronological order of the term.

9. The student number and name of each fourth year student, together with three measures: the average grade of the student's completed CS courses, the average grade of all completed courses by the student, and the percentage of all completed courses that were CS courses by the student. The result should be sorted in descending order of the first measure, and then by descending order of the third measure.

10. The percentage of professors who have never taught in the past, during the same term, at least two classes for two different courses, and for whom this is not the case with the current term. Note that a percentage should be a number between 0 and 100. (You can assume there is at least one professor.)