

# CS 457 / CS 557 – Database Software Design

## Assignment 3

### Instructions

This assignment is split into two problems. The first problem concerns the practice of SQL queries to interrogate and to update a database. The second problem involves questions that require reading about *File Structures, Indexing, and Hashing*. Please, proceed as follow:

- Be sure to terminate every SQL statement with semicolon ";".
- Using your favorite editor, create *Assignment3P1.sql* that contains the queries of the problem 1.
- Include your name (and the name if your teammates if it is applicable) in a header comment at the top of your source file.
- Make sure that the output of each query is distinguishable. Comment your code; if nothing else, mark each query with its number.
- The answers of the problem 2 must be submitted in pdf file.
- Reformulate your answers of the problem 2 to avoid plagiarism.
- Good luck ☺.

### Problem 1

Write the following queries in SQL, using the university schema from the *assignment 2*.

1. Find the names of all students who have taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.
2. Find the IDs and names of all students who have not taken any course offering before Spring 2009.
3. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.
4. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.
5. Create a new course “CS-001”, titled “Weekly Seminar”, with 0 credits.
6. Create a section of this course in Autumn 2009, with sec id of 1.
7. Enroll every student in the Comp. Sci. department in the above section.

8. Delete enrollments in the above section where the student's name is Chavez.
9. Delete the course CS-001. What will happen if you run this delete statement without first deleting offerings (sections) of this course?
10. Delete all takes tuples corresponding to any section of any course with the word "database" as a part of the title; ignore case when matching the word with the title.
11. Suppose that we have a relation *marks*(*ID*, *score*) and we wish to assign grades to students based on the score as follows: grade F if  $score < 40$ , grade C if  $40 \leq score < 60$ , grade B if  $60 \leq score < 80$ , and grade A if  $80 \leq score$ . Write SQL queries to do the following:
  - a. Display the grade for each student, based on the marks relation.
  - b. Find the number of students with each grade.
12. The SQL like operator is case sensitive, but the *lower()* function on strings can be used to perform case insensitive matching. To show how, write a query that finds departments whose names contain the string "sci" as a substring, regardless of the case.

### **Problem 2: reading exercise**

1. What is the difference between primary and secondary storage?
2. Why are disks, not tapes, used to store online database files?
3. Discuss the process of disk initialization.
4. Why is accessing a disk block expensive? Discuss the time components involved in accessing a disk block.
5. What are the reasons for having variable-length records? What types of separator characters are needed for each?
6. Discuss the techniques for allocating file blocks on disk.
7. What is the difference between a file organization and an access method?
8. What is the difference between static and dynamic files?

Recommended book: R. Elmasri, S. B. Navathe. Fundamentals of Database Systems. 6<sup>th</sup> Edition, Addison-Wesley, 2011. (Available on Moodle in suggested eBooks).