

CS411 Database Systems

Spring 2007

HW #2 - Part 2

Due: 1:55pm CST, 03/06/07

Note: Print your name and NetID in the upper right corner of every page of your submission. Handin your stapled homework to Trisha Benson in 4322 SC. In case Trisha is not in office, slide your homework under the door.

Problem 5 SQL

Consider the following relations:

Students(sid, sname, department)

Courses(cid, cname, department)

Register(sid, cid, semester, year, grade)

Express the following queries in SQL.

(5.1) Find the names of all CS students;

(5.2) For each course, show its total number of registrations in “Spring” semester of 2006;

(5.3) Find the courses with the highest total number of students from departments different from where the course is offered;

(5.4) Find the names of students who have taken courses in both CS and ECE;

(5.5) Find the records of A+ in CS courses. For each such record, return the student name, course name, semester, and year.;

Problem 6 View

Continue Problem 5,

(6.1) Create a view **LoyalStudent** that records every pair of student and department if that student has taken more than 10 courses from that department. You can include as many columns (attributes) in the view as necessary, as long as the view can be used in answering the following question.

(6.2) Using the view **LoyalStudent**, find the CS courses that have never been taken by any “loyal student” of CS courses. Return the course names.

Problem 7 Definition and Modification

With the following relations, write the following SQL declarations and modifications.

Students(sid, sname, department)

Courses(cid, cname, department)

Register(sid, cid, semester, year, grade)

- (7.1) Define a suitable schema for relation **Students**, **Courses**, and **Register**, respectively; Declare the keys on **Students**, **Courses**, and **Register**;
- (7.2) Insert into the database the fact that ECE student “John Smith”, with id “1001”, got *A* in CS 101 (*i.e.*, course id *cid* = 101) in Spring 2006;
- (7.3) Delete all courses in ”General Engineering” that do not have any students registered.

Problem 8 Constraints and Triggers

Consider the following relational schema.

Students(sid, sname, department)

Courses(cid, cname, department)

Register(sid, cid, semester, year, grade)

- (8.1) Suppose you want to enforce the constraint that no registration year is later than 2007.
- (a) Can you use *CHECK* to implement this constraint? If Yes, how (provide SQL)? If No, why?
- (b) Answer the same questions for *assertion* instead of *CHECK*.
- (c) Answer the same questions for *trigger* instead of *CHECK*.
- (8.2) Suppose you want to enforce the following constraint. When inserting a tuple into the **Register** table, *sid* of the inserted tuple must exist in the *sid* column of table **Students**. If not exist, then it must be automatically inserted into the table **Students**.
- (a) Can you use *CHECK* to implement this constraint? If Yes, how (provide SQL)? If No, why?
- (b) Answer the same questions for *assertion* instead of *CHECK*.
- (c) Answer the same questions for *trigger* instead of *CHECK*.