CS54100 - Homework 1
Fall 2016

Due: Friday, September 23 in class, before class starts.
(There will be a 10% penalty for each late day. After 5 late days, the homework will not be accepted.)

Part 1. E-R Diagram (30 points)

1. (20 points)

You are asked to design the E-R diagram of an airport database system. The system manages planes and employees at an airport. State any assumptions that you make. Make sure that your design captures the following aspects:

A. Airplanes:
   Each airplane has a unique registration number and is of a specific model.

B. Airplane models:
   Each model has a model number as well as a capacity and weight.

C. Employees:
   All employees at the airport can be identified by a unique social security number. In addition, all employees belong to a union and have a unique union membership number. There are two types of employees that the database needs to track: Technicians and Traffic Controllers.
   Traffic Controllers are required to take medical exams annually. The database must capture the latest medical examination date.
   The database needs to store the name, phone number, and salary of all technicians. In addition, each technician has expertise in servicing one or more airplane models that must be tracked. Multiple technicians can have expertise in the same type of model.

D. Airplane tests:
   Tests must be administered periodically to each airplane to determine if they are fit for service. For each test, the database must store the Federal Aviation Administration (FAA) test number, test name, and max score possible on the test.
   The database must also store the test results for each test performed on each airplane. The database needs to track when an airplane is tested by a given technician using a given test. The database needs to store the date the test was performed, the number of hours the technician spent on the test, and the score the airplane received on the test.

2. (10 points)

Convert the E-R model you built into the corresponding relational model. You do not need to use formal CREATE TABLE statements to specify the relational model. Just list the schemas, making sure to include any referential integrity constraints and any other constraints that are necessary. State any assumptions you make.

Part 2. Relational Algebra and Calculus (30 points)

Consider the **CourseRegistry** relational schema given in Project 1. Write the following queries in Relational Algebra (Questions 1,3,5,7,9, below), Query-By-Example (Questions 2,4,6,8,10, below), Tuple Relational Calculus (Questions 1,3,5,7,9, below), and Domain Relational Calculus (Questions 2,4,6,8,10, below). Note that some of these queries might not be expressible in Relational Algebra, QBE, or Relational Calculus. For such queries, informally explain why they cannot be expressed.

A detailed definition of TRC is available in the "Database Management Systems" textbook.

1. Find the names of students who are not enrolled in any course.
2. Find the names of students who are not enrolled in more than three courses.
3. Find the courses taught by faculty from more than two departments.
4. Find the name of the course with the highest total maximum occupancy (over all course offerings).
5. Find the class standing with the second-highest average GPA.
6. Find the students who have taken courses only with one faculty.
7. Find the courses that have been taken by all students.
8. Find students with a GPA > 3.0 that are taking exactly two courses.
9. Find courses taught by only one faculty member.
10. Find courses with at least one student from every major currently in the database.

Part 3. SQL (10 points)

Consider the **CourseRegistry** relational schema given in Project 1.

Given the nested queries below,

A. State what each query computes, and
B. State, if any, the equivalent un-nested (single) SQL query.

1.

```
SELECT C.cname
FROM Courses C
WHERE C.credits >= 3.0
      EXISTS (
            SELECT *
            FROM Offerings O, Faculty F
            WHERE F.deptid = 5 AND
                  F.fid = O.fid AND
                  O.cnum = C.cnum
         );
```

2.

```
SELECT C.cname
FROM Courses C
WHERE C.cnum NOT IN (
            SELECT O.cnum
            FROM Offerings O, Enrolled E, Students S
            WHERE O.onum = E.onum AND
                  E.snum = S.snum AND
                  (S.standing = 'FR' OR S.standing = 'SO')
      );
```