CIS 717.1 Final Exam

You get $\frac{1}{4}$ points if you leave an answer blank. You don't get points for a totally wrong answer. You *might* get partial credit for a partially correct answer.

1. (10 points) Describe *two different* ways of implementing one-to-one relationships. Assume we are maintaining information on offices (office number, building, phone number) and faculty (number, name, etc.). No office houses more than one faculty member. No faculty member is assigned more than one office. Illustrate the ways of implementing one-to-one relationships using offices and faculty. Which approach would be best in each of the following situations?

   (a) A faculty member must have an office and each office must be occupied by a faculty member.

   (b) A faculty member must have an office, but some offices are not currently occupied. (We still need to maintain information about these unoccupied offices.)

   (c) Some faculty members do not have an office. All offices are occupied, however.

   (d) Some faculty members do not have an office. Some offices are not occupied.

2. (10 points) Construct an E-R diagram for a school that has a set of classes, sections, faculty, and students. Classes may have zero or more offered sections. Each sections needs to be taught by a faculty member. A student may register for many sections. A section may have many students registered for it. *Document all assumptions that you make about the mapping constraints.*

   Be sure to address the following: How does a student register? How to retrieve student's transcript? How to retrieve faculty and student schedules?

3. (10 points) Repeat previous question only using UML.

4. (10 points) List "objects" and "events" in your design (from above questions). Which one is an "object"? Which one is an "event"? Why? Write `CREATE TABLE` statements for one object and one event.

5. (10 points) For the following, use:

   ```
   employee(employee-name, street, city)
   works(employee-name, company-name, salary)
   company(company-name, city)
   manages(employee-name, manager-name)
   ```

   Write SQL code to answer each question:

   (a) Find 'John Doe's Manager's Name.

   (b) Find all employees in the database who live in the same cities and on the same streets as do their managers.

   (c) Find all employees who earn more than the average salary of all employees in their company.

   (d) Find the company that has the smallest payroll.

   (e) Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.

6. (10 points) Use the same schema as in previous question.

   (a) Using relational algebra: Find 'John Doe's Manager's Name.

   (b) Using relational algebra: Find all employees in the database who live in the same cities and on the same streets as do their managers.

7. (10 points) For the following, use:

branch-name	*loan-number*	*amount*
Downtown	L-170	3000
Redwood	L-230	4000
Rerryridge	L-260	1700

loan

customer-name	*loan-number*
Jones	L-170
Smith	L-230
Hayes	L-155

borrower

   (a) Show result of "loan inner join borrower".

   (b) Show result of "loan left outer join borrower".

   (c) Show "loan natural inner join borrower".

   (d) Show "loan natural right outer join borrower".

   (e) Show "loan full outer join borrower using (loan-number)".

8. (10 points) In *your own* words, describe what it means for a database to be in:

   (a) First-Normal Form.

   (b) Second-Normal Form.

   (c) Third-Normal Form.

   (d) Boyce-Codd Normal Form.

9. (10 points) For each normal form from the previous question, describe the positive and negative benefits that your database/application gets. i.e.: Exactly what your database gains/loses when it is in second-normal form? How about third-normal form?, etc.

10. (10 points) Modify the database in question 5 to keep track of employee salary history. For example, you would like to know the dates when the employee got a raise, and what was their previous salary before those raises. (you can draw a UML diagram, or write create table statements, as the answer)

    What are the implications of such a change? How will someone find employee's current salary? (write a SQL query for that). Using stored procedures or triggers, how can you simplify things?

Good luck!