CSE 562: Homework #1 (due 02/20/12)

Submission

Submit all the answers using `submit_cse562` as a **single pdf file**.

This is **individual work**. Copied solutions will be considered violations of academic integrity. Please write your name and the text

> **"The submitted solutions are my individual work."**

at the beginning of the submitted file.

Database

You are given the following relational schema:

```
STUDENT(SNO,SNAME,DEPT)
ENROLL(CNO,SNO,GRADE)
COURSE(CNO,DEPT)
PREREQ(CNO,PNO)
```

Keys are underlined. The column `ENROLL(SNO)` is a foreign key referencing `STUDENT(SNO)`. All the occurrences of the columns `CNO` and `PNO`, except for the one in `COURSE`, are foreign keys referencing `COURSE(CNO)`.

Course numbers are strings, like *CSE562*. The column `COURSE(DEPT)` contains just the departmental acronym, e.g., `CSE`.

You can assume that no columns are null. You may use views.

The example database is available in the following tables: `zhouhany.student`, `zhouhany.enroll`, `zhouhany.course` and `zhouhany.prereq`.

Problem 1 (12 pts)

Write the following queries in relational algebra and SQL:

- **1.1**: *Find the names of all the students enrolled in CSE562;*
- **1.2**: *Find the names of all the students, whose grade in CSE531 is lower than the grade in CSE562, or who took at most one of CSE531 and CSE562;*
- **1.3**: *Find all the EE students who took all the courses offered by CSE.*

Problem 2 (8 pts)

Write the following queries in SQL:

- **2.1**: *For every course, list the course together with the average grade in that course;*
- **2.2**: *For every course, list the course together with the department that had the maximum number of students in that course. If multiple departments are in that category, list all of them. If a course has zero enrollment, return null instead of the department name.*

Problem 3 (8 pts)

Write the following queries in SQL, possibly using recursion:

- **3.1**: *Find all the prerequisites, direct or indirect, of CSE562, assuming that the maximum length of prerequisite chains is 3.*
- **3.2**: *Find all the prerequisites, direct or indirect, of CSE562, without assuming a maximum length of prerequisite chains.*
- **3.3**: *Find all the departments that have a course requiring a CSE course as a direct or indirect prerequisite, without assuming a maximum length of prerequisite chains.*

Note: Please bear in mind that Oracle does not support recursion. If you want to test your recursive queries, use PostgreSQL.

Problem 4 (12 pts)

Consider the following SQL queries:

- **4.1**:

```
SELECT *
FROM STUDENT S
WHERE NOT EXISTS
  SELECT * FROM ENROLL E
  WHERE E.SNO = S.SNO
    AND E.GRADE > 3.0
```

- **4.2**:

```
SELECT E1.CNO, S.SNAME
FROM ENROLL E1, STUDENT S
WHERE E1.SNO = S.SNO
  AND NOT EXISTS
    SELECT * FROM ENROLL E2
    WHERE E2.CNO = E1.CNO
      AND E2.GRADE > E1.GRADE
```

For both queries:

- Explain what the query is doing.
- Translate the query to relational algebra.

Problem 5 (10 pts)

The *relax-join* operator is defined as follows:

- if the natural join of the relations R and S is nonempty, then return the result of this join;
- otherwise, return the Cartesian product of R and S.

Write queries in relational algebra and SQL that return the relax-join of two relations. Do not use IF-THEN-ELSE.

Update (02/13/12). It is enough to consider natural joins defined as:

$$\pi_{A,B,D}(R(A,B) \underset{B=C}{\bowtie} S(C,D)).$$

Update (02/16/12). Replace "natural join" by "equijoin." Replace the above expression by

$$R(A,B) \underset{B=C}{\bowtie} S(C,D).$$