CS145 Written Assignment #8

NOTE: *You do not need to turn anything in for this assignment. The sample solution has already been posted on the course Web page. It should help you prepare for the final exam.*

1. Suppose we have a relation $R(A, B, C, D)$ with FD's $A \to C$, $D \to C$, and $BD \to A$. Show that the decomposition of R into $R_1(A, B)$, $R_2(A, C, D)$, and $R_3(B, C, D)$ is not lossless.

   Note: In Lecture Notes #14, we have only defined the lossless decomposition into two relations. It is quite straightforward to extend the definition to more relations. In general, if a relation R has a set of FD's $\mathcal{F}$, and R is decomposed into $R_1, R_2, ..., R_k$, we say that the decomposition is a *lossless (join) decomposition (with respect to $\mathcal{F}$)* if for every instance of R satisfying $\mathcal{F}$:

   $$R = \pi_{attrs(R_1)}(R) \bowtie \pi_{attrs(R_2)}(R) \bowtie ... \bowtie \pi_{attrs(R_k)}(R)$$

   That is, R is the natural join of its projections.

2. Consider a relation $R(A, B, C, D)$ with a set of FD's $\mathcal{F} = \{AB \to C, A \to D, BD \to C\}$.

   (a) Find all keys of R.

   (b) Decompose R into BCNF, starting with the BCNF violation $A \to D$.

   (c) Show that the decomposition in (b) does not preserve all FD's in $\mathcal{F}$ (i.e., some FD in $\mathcal{F}$ does not follow from the FD's that hold in the decomposed relations).

   (d) Find $\mathcal{F}_c$, a canonical cover for $\mathcal{F}$.

   (e) Decompose R into 3NF.

   (f) Show that the decomposition in (d) preserves all FD's in $\mathcal{F}$ (i.e., every FD in $\mathcal{F}$ follows from the FD's that hold in the decomposed relations).

3. Prove (using the definition of MVD) or disprove (by giving a counterexample) the following rules for MVD's:

   (a) Combining: if $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$, then $X \twoheadrightarrow YZ$

   (b) Splitting: if $X \twoheadrightarrow YZ$, then $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$

4. Let us consider the relation schema $CTHRSG$, where C = course, T = teacher, H = hour, R = room, S = student, and G = grade. The dependencies we assume are:

   - $C \to T$: Each course has one teacher.
   - $HR \to C$: Only one course can meet in a room at one time.
   - $HT \to R$: A teacher can be in only one room at one time.
   - $CS \to G$: Each student has one grade in each course.
   - $HS \to R$: A student can be in only one room at one time.
   - $C \twoheadrightarrow HR$: Given a course, there is an associated set of hour-room pairs, namely the meeting times and locations of the course.

(a) Show that $HR \twoheadrightarrow SG$ follows from the given set of dependencies. What does $HR \twoheadrightarrow SG$ mean in English?

(b) Find all keys of $CTHRSG$.

(c) List all BCNF violations.

(d) List all 4NF violations.

(e) Decompose $CTHRSG$ into 4NF.

5. Consider the following ODL schema for a database of students applying for summer internships.

```
interface Student (extent Students, key ID) {
    attribute integer ID;
    attribute Struct{string first, string last} name;
    relationship Set<Internship> applied
        inverse Internship::applicants;
};
interface Internship (extent Positions, key (company, city)) {
    attribute string company;
    attribute string city;
    relationship Set<Student> applicants
        inverse Student::applied;
};
```

Write OQL queries for each of the following.

(a) Find the ID's of all students whose last name is Simpson.

(b) Find the ID's and last names of all students who have applied to an internship at a company in Palo Alto. Do not repeat (ID, last-name) pairs in the result, even if the student has applied to many internships in Palo Alto.

(c) If you used `DISTINCT` in your answer for part (b), rewrite the query so you don't need to use `DISTINCT`. Conversely, if you didn't use `DISTINCT` in your answer for part (b), rewrite the query so you do need to use `DISTINCT` in order to guarantee that duplicates are eliminated.

(d) Find the names of all companies in Palo Alto such that at least one student s (say) with ID between 100 and 300 has applied for an internship at that company, and all internships student s has applied for are in Palo Alto or San Jose.

(e) Recall that the result of an OQL query or subquery is a set or a bag. OQL allows two sets (bags) to be compared using $=$, where two sets (bags) are equal if they contain exactly the same objects. Find all pairs of student ID's such that the two students have applied to internships at the exact same set of companies in Palo Alto. (The students may have applied to different internships at companies in other cities.) Return each pair of ID's exactly once, and order the final result based on the last name of the first student in each pair.

(f) Can you write the query in part (e) without using set or bag equality? If so, write it. If not, explain why not.