Written Assignment #3
Due Wednesday April 29

1. Consider a database for a university that may include information about students, courses, professors, etc.

   (a) Specify an example relation for this database and a set of functional dependencies over the relation such that the relation is in Third Normal Form (3NF) but is not in Boyce-Codd Normal Form (BCNF). Your relation need not be extensive (i.e., it need not include many attributes and it may represent only a very small part of the complete university information), but the schema and the dependencies should be realistic in their modeling of the real world. You should make no assumptions besides those captured by the schema and the functional dependencies.

   (b) Decompose your relation from part (a) so that the new schema is in BCNF. Are there any functional dependencies that cannot be checked in the decomposed relations, assuming that dependencies are checked on one relation at a time?

   (c) Now specify an example relation for this database and a set of functional and multivalued dependencies over the relation such that the relation is in BCNF but is not in Fourth Normal Form (4NF). Again, your relation need not be extensive but the schema and the dependencies should be realistic in their modeling of the real world, and you should make no assumptions besides those captured by the schema and the dependencies. The relation you give for this part of the problem can capture similar or different information from that captured in part (a).

   (d) Decompose your relation from part (c) so that the new schema is in 4NF. Are there any functional or multivalued dependencies that cannot be checked in the decomposed relations, assuming that dependencies are checked on one relation at a time?

2. There is a rule (page 160 of the textbook) that states "every functional dependency is a multivalued dependency." Show that the converse is not true—it is not true that every multivalued dependency is a functional dependency. Show this by providing a counterexample: Give a relation schema, a multivalued dependency $A_1, \ldots, A_n \twoheadrightarrow B_1, \ldots, B_m$, and a relation instance (set of tuples) for which the multivalued dependency holds but the corresponding functional dependency $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ does not hold. As an additional requirement, the multivalued dependency in your counterexample should be a nontrivial one, but otherwise please give the simplest counterexample you can come up with.

3. Prove the correctness of the *complementation rule* for multivalued dependencies. The definition of the rule is given on page 160 of the textbook. Your proof should be based on the formal definition of multivalued dependency (bottom of page 157) and should have the general form: "Suppose $A_1, \ldots, A_n \twoheadrightarrow B_1, \ldots, B_m$ holds. Then for each pair of tuples t and u, $\ldots\langle$fill in this part$\rangle \ldots$ Therefore $A_1, \ldots, A_n \twoheadrightarrow C_1, \ldots, C_k$ holds."

(over)

4. See Exercise 4.1.1 on pages 187–190 of the textbook.

(a)–(g) For each of parts (a) through (g) in Exercise 4.1.1, write a relational algebra expression for the query and show the result of the query on the data of Figures 4.9 and 4.10.

(h)–(k) **Optional:** If you have time on your hands and want to impress the TA, also write queries and show results for parts (h) through (k) in Exercise 4.1.1.

5. (a) The *semijoin* of relations R and S, denoted $R \ltimes S$, returns all tuples in R that agree with at least one tuple in S on the attributes in $schema(R) \cap schema(S)$. Rewrite $R \ltimes S$ using standard relational operators.

   (b) The *antisemijoin* of relations R and S, denoted $R \mathbin{\overline{\ltimes}} S$, returns all tuples in R that do not agree with any tuples in S on the attributes in $schema(R) \cap schema(S)$. Rewrite $R \mathbin{\overline{\ltimes}} S$ using standard relational operators.

   (c) The six relational operators σ, π, $\times$, $\cup$, $-$, and ρ form an *independent set*, meaning that none of the operators in the set can be defined in terms of the other five. (See Section 4.1.9 in the textbook for more discussion.) If we decide to include $\overline{\ltimes}$ in this set, which operation should we remove in order to keep the set independent? Briefly justify your answer.

6. Consider three relations $R(A, B)$, $S(B, C)$, and $T(C, A)$. Suppose that each relation contains n tuples. For each of the following expressions, give the minimum number of tuples (as a function of n) that the result of the expression could have. Also state briefly or give a concrete example of how the minimum is achieved.

   (a) $R \bowtie S$

   (b) $R \bowtie S$ if we know that $R = R \ltimes S$ and $S = S \ltimes R$ ($\ltimes$ is defined in Problem #5)

   (c) $R \bowtie S \bowtie T$

   (d) $R \bowtie S \bowtie T$ if we know that:

   $$R = R \ltimes S$$
   $$R = R \ltimes T$$
   $$S = S \ltimes R$$
   $$S = S \ltimes T$$
   $$T = T \ltimes R$$
   $$T = T \ltimes S$$

   Hint for part (d): The answer for $n = 1$ is different from the answer for $n \neq 1$.

7. **Personal Database Application (PDA)**

   Remind us of your final schema from Problem #8(d) in Written Assignment #2, and the set of functional dependencies for your application.

   (a) If you had decomposed your relations based on 3NF instead of BCNF, would you have gotten the same result?

   (b) Are there any nontrivial multivalued dependencies that hold on any of the relations in your schema? If so, decompose the relations into smaller ones so that each one is in 4NF.

   (c) Is there anything you still don't like about the schema? This is your last chance to fiddle with it on paper before committing it to bits in Oracle.