# Written Assignment #2
# Due Wednesday April 22

1. Consider your solution to Problem #2 on Written Assignment #1. Please redraw your E/R diagram or specify your ODL schema again.
   (a) Using the method for translating an E/R diagram or ODL schema to relations, produce a set of relations for your database design. Please be sure to underline key attributes in the relations. In cases where we gave alternative mappings for ODL constructs (such as for sets and relationships), you may use whichever mapping you prefer.
   (b) Give a different relational schema from the one you got in part (a) that still captures the same information.

2. Consider your solution to Problem #1 on Written Assignment #1. If you used an E/R diagram for Problem #1 on this assignment, then consider your ODL schema for Problem #1 on Written Assignment #1. Otherwise, if you used an ODL schema for Problem #1 on this assignment, then consider your E/R diagram for Problem #1 on Written Assignment #1. Please redraw your E/R diagram or specify your ODL schema again.
   (a) Using the method for translating an E/R diagram or ODL schema to relations, produce a set of relations for your database design. Please be sure to underline key attributes in the relations. In cases where we gave alternative mappings for ODL constructs (such as for sets and relationships), you may use whichever mapping you prefer.
   (b) Give a different relational schema from the one you got in part (a) that still captures the same information.

3. Consider the four relations on pages 188–189 of the Ullman/Widom textbook. These relations contain a database for describing PC's, laptops, and printers.
   (a) Reverse-engineer a corresponding E/R diagram for this relational schema. That is, give an E/R diagram that would be mapped to this set of relations.
   (b) Give an alternate relational design for this database that uses only one relation.
   (c) Briefly discuss the advantages and disadvantages of the original four-relation design versus the one-relation design in part (b).
   (d) Consider again the original four relations. Notice that *price* is an attribute in each of the `PC`, `Laptop`, and `Printer` relations. Can we remove it from these three relations and instead add a *price* attribute to the `Product` relation? If yes, what are the advantages or disadvantages of doing so? If no, why not?

4. Consider a relation $R(A, B)$ where attributes $A$ and $B$ contain values that are bit strings. The instance of $R$ with four tuples shown below satisfies the functional dependency $A \rightarrow B$ and all functional dependencies that follow from $A \rightarrow B$, but it does not satisfy any other functional dependencies (in particular, it does not satisfy $B \rightarrow A$):

| $A$ | $B$ |
|---|---|
| 00 | 0 |
| 01 | 0 |
| 10 | 1 |
| 11 | 1 |

This example serves as a hint for parts (a) and (b) that follow.

(a) Consider a relation $R(A, B, C)$ where all attributes contain values that are bit strings. Find an instance of $R$ that satisfies the functional dependencies $A \rightarrow B$, $B \rightarrow C$, and all functional dependencies that follow from these two, but it does not satisfy any other functional dependencies (such as $C \rightarrow A$, $BC \rightarrow A$, etc.).

(b) Consider a relation $R(A, B, C, D)$ where all attributes contain values that are bit strings. Find an instance of $R$ that satisfies the functional dependencies $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, and all functional dependencies that follow from these three, but it does not satisfy any other functional dependencies (such as $C \rightarrow A$, $D \rightarrow B$, $CD \rightarrow A$, etc.).

5. Consider a relation $R(A, B, C, D, E)$. Suppose that the following five functional dependencies hold on $R$:

$$\begin{array}{rcl} A & \rightarrow & D \\ AB & \rightarrow & C \\ B & \rightarrow & E \\ D & \rightarrow & C \\ E & \rightarrow & A \end{array}$$

Now suppose that we decompose relation $R$ so that one of the new relations is $R1(A, B, C)$. Given the complete set of functional dependencies above, specify all keys for $R1$. Don't forget that a key must be *minimal*.

6. Consider the following "rule" for functional dependencies:

If $A \rightarrow B$ and $BC \rightarrow D$ then $AC \rightarrow D$

(a) Is this "rule" correct or incorrect?

(b) If you said in part (a) that the rule is correct, sketch a simple proof based on the formal definition of functional dependencies. If you said in part (a) that the rule is incorrect, give a simple counterexample that includes a relational schema and a set of tuples that satisfy the "if" part of the rule but not the "then" part.

7. A database designer has as his first assignment to design the schema for a company database. Each employee has an ID (unique across employees), Name, Address, Office, and Salary. The designer decides to create the following four relations:

```
EmpName(ID, Name)
EmpAddress(ID, Address)
EmpOffice(ID, Office)
EmpSalary(ID, Salary)
```

(a) State the completely nontrivial functional dependencies for each relation.

(b) Are all four relations in Boyce-Codd Normal Form?

(c) Is this a good database design? Why or why not?

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

   **⇒ Please make yourself a copy of your solution to this problem before turning it in. You will need it for Written Assignment #3.**

   (a) Consider the E/R diagram or ODL schema you designed for your PDA in Problem #6 of Assignment #1. Please redraw your E/R diagram or specify your ODL schema again. Using the method for translating an E/R diagram or ODL schema to relations, produce a set of relations for your database design. As usual, please be sure to underline key attributes in your relations, and in cases where we gave alternative mappings for ODL constructs (such as for sets and relationships), you may use whichever mapping you prefer.

   (b) For each relation in the schema produced in part (a), specify a set of completely nontrivial functional dependencies for the relation. Any functional dependencies that actually hold in the real-world scenario that you're modeling should be specified, or should follow from the specified dependencies. Don't worry if you find that some of your relations have no nontrivial functional dependencies.

   (c) Is each relation in your schema in Boyce-Codd Normal Form with respect to the functional dependencies you specified in part (b)? If not, decompose the relation into smaller relations so that each relation is in BCNF.

   (d) Is there anything you still don't like about the schema (e.g., attribute names, relation structure, etc.)? If so, modify the relational schema to something you prefer. You will be working with this schema quite a bit, so it's worth spending some time to make sure you're happy with it.