Assignment 6: Normal Forms and Functional Dependencies

Important Collab-Policy Note!

You are allowed to use computer programs to solve the functional dependency problems, as long as they are programs that you write entirely by yourself. You cannot use programs written by other students, or from other sources, to compute the results of the functional dependency problems.

- Although it is tons of fun to write programs that analyze functional dependencies, historically it has been faster for students to do the problems by hand, rather than write programs and then compute the answers. So you have been warned…
- If you do write a program to solve any of the functional dependency problems, you must turn in the program as well.

Additional Note

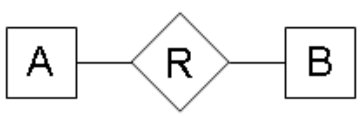
Just as a point of clarity, none of the problems in this assignment require you to write SQL. Some problems ask you to create relation schemas and specify their primary/candidate/foreign keys. You can do this using the relational model schema notation we have used all term, with additional notes listing the candidate and foreign keys. You don't have to create any SQL files though.

Problems

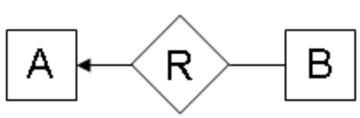
1. Here are four simple E-R diagrams. The entity-set schemas are $A(a)$ and $B(b)$, and R will obviously have the schema (a, b). For each diagram, specify what the non-trivial functional dependencies are, if any. If there are no non-trivial functional dependencies, make sure you state that.

   (2 pts per part; 8 pts total)

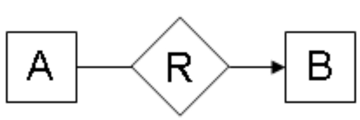
   a) Many-to-many mapping between A and B



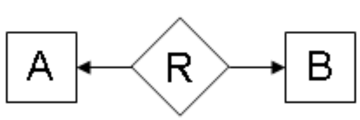
   b) One-to-many mapping between A and B



   c) Many-to-one mapping between A and B



   d) One-to-one mapping between A and B



2. For the Union rule, the Decomposition rule, and the Pseudotransitivity rule (pg. 339 in the textbook), use Armstrong's axioms to prove that each of these rules is sound. Make sure to specify what axiom(s) you apply for each step of your proofs. *(3 pts per rule; 9 pts total)*

3. Given the relation schema $R = (A, B, C, D, E)$, and this set of functional dependencies:

   $F = \{ A \to BC, CD \to E, B \to D, E \to A \}$

   a) Specify all candidate keys for R. You do not need to prove exhaustively that they are candidate keys, but make sure to show that each candidate key is a superkey for R.

   b) Describe <u>all</u> functional dependencies that will appear in the closure F^+ of F. **This includes dependencies inferred from the above set, as well as trivial functional dependencies.** You can either describe F^+ by computing the entire set exhaustively (e.g. with a program you write yourself), or you can summarize the dependencies:
      - For example, you might write: "$ABCDE \to ABCDE$, and all dependencies generated from this by applying the Decomposition Rule."
      - You might also use variables in your summarizations, such as: "All trivial dependencies $\alpha \to \beta$ where $\alpha = \{ ABCDE \}$ and $\beta \subseteq \alpha$." This is a simple example; you can also do more involved things along these lines, e.g. involving the candidate keys you find.

      Just be sure to clearly indicate <u>exactly</u> what you are summarizing, so that it's clear that your answer includes the entire closure of F.

      (Note: It tends to be faster/easier to simply summarize the dependencies than to write the program, but writing a program is pretty fun. Just expect it to take some time to write…)

   (10 points total)

4. Given a relational schema $R(A, B, C, D)$, does $A \twoheadrightarrow BC$ logically imply $A \twoheadrightarrow B$ and $A \twoheadrightarrow C$? If yes then prove it, otherwise give a counterexample. *(5 points)*

   (**Aside:** The double arrow "$\to\to$" is often used to indicate a multivalued dependency.)

5. Given a relation-schema $R(A, B, C, D, E, G)$, and the set of functional dependencies:

   $F = \{ A \to E, BC \to D, C \to A, AB \to D, D \to G, BC \to E, D \to E, BC \to A \}$

   a) Compute a canonical cover F_c of F. Show each step of your work:

      - If you use inference rules on a step then note which rules on the step.
      - If you assert that a particular attribute is extraneous in a dependency, show a proof that this is the case.

      Failure to record the details of each step will result in point deductions.

   b) Find a candidate key for R. Demonstrate that it is a superkey for R by computing its attribute-set closure. Then, demonstrate that it is a candidate key by computing the attribute-set closures of its proper subsets.

c) Decompose R into a BCNF schema. Prove that each of your final relation-schemas is indeed in BCNF. (You don't have to do this exhaustively, but make sure that you clearly demonstrate that each relation-schema is indeed in BCNF.)

Note all dependencies in F_c that are not preserved by this BCNF decomposition.

d) Find a second BCNF decomposition for R. (Hint: when you choose how to decompose a particular schema, you sometimes have multiple choices.) Again, prove that each of the final relation-schemas is indeed in BCNF. (If you already proved that a particular schema is in BCNF in part c, just write a note like, "R_i is in BCNF; see part c.") As before, note all dependencies in F_c that are not preserved by this BCNF decomposition.

e) Using the 3NF schema synthesis algorithm, create a 3NF schema for R.

(Part a: 8 pts; part b: 5 pts; part c: 9 pts; part d: 9 pts; part e: 7 pts. Total: 38 pts.)

6. As we saw in the previous problem, a particular schema can be normalized in several different ways, even when using only a single normal form. What actually makes the most sense depends on the enterprise that the schema is actually modeling.

Consider this schema for a database of courses and sections: R(*course_id*, *section_id*, *dept*, *units*, *course_level*, *instructor_id*, *term*, *year*, *meet_time*, *room*, *num_students*). The department, term and year are included, as is the instructor's ID, and the details of when and where each course is held. Of course, this schema is not normalized at all.

These functional dependencies also hold on R:

{ *course_id* } → { *dept*, *units*, *course_level* }
{ *course_id*, *section_id*, *term*, *year* } → { *meet_time*, *room*, *num_students*, *instructor_id* }
{ *room*, *meet_time*, *term*, *year* } → { *instructor_id*, *course_id*, *section_id* }

a) Find all candidate keys of R. Prove that each one is a superkey. (You don't have to prove that they are candidate keys; that would be annoying.)

b) Identify all canonical covers F_c for the above set of functional dependencies. *(Hint: there are two.)* Since there are multiple options for F_c, identify the F_c that is *most appropriate* for what is being modeled by the schema and dependencies above. Explain your rationale.

c) Suggest both a normal form and a schema decomposition that would be best for actual use in a course management system. Briefly explain your rationale. Make sure to indicate any and all primary, candidate, and foreign keys in your answer.

You should take into account both the scale of this database (i.e. how many records would be managed in such a database, and the practicality or impracticality of enforcing constraints in this database), as well as the need for ensuring a correct and complete representation in the database.

(Part a: 6 pts; part b: 8 pts; part c: 6 pts. Total: 20 pts.)

7. Given the following database, and set of functional and multivalued dependencies:

   emails(*email_id*, *send_date*, *from_addr*, *to_addr*, *subject*, *email_body*,
   attachment_name, *attachment_body*)

   $email\_id \rightarrow send\_date, from\_addr, subject, email\_body$
   $email\_id \twoheadrightarrow to\_addr$ (a multivalued dependency)
   $email\_id, attachment\_name \rightarrow attachment\_body$

   Create a set of 4NF relation schemas for the above database. Make sure to indicate all primary and foreign keys in your resulting schemas. Give each of your resulting schemas a descriptive name; don't just call them R_1, R_2, etc.

   Also, for each of your resulting schemas, demonstrate that each schema is in 4NF by citing the appropriate criteria from the 4NF normal form conditions, as well as indicating which of the above dependencies are relevant to your statements.

   (10 points)