Databases Homework 4

1. Below are two relations. Each relation has a corresponding set of FDs that hold in the relation. For each relation and corresponding FDs, answer these three questions:

   (i) What are all the nontrivial FDs that follow from the given FDs? List each FD with a single attribute on the right side.

   (ii) What are the keys of the relation?

   (iii) What are the superkeys for the relation that are not keys?

   Schemas and FDs:

   (a) $R(A, B, C, D)$ with FDs $C \to B$, $C \to A$, and $AB \to D$.

   (b) $R(A, B, C, D)$ with FDs $AB \to C$, $C \to D$, and $D \to B$.

2. For all the parts of this question, only write FDs which are completely nontrivial and which have single attributes on the right side. Also, in each question, you only need to provide a minimal basis of FDs (though if you give extras, that's OK).

   (a) Suppose we have a relation PizzaOrders(customerId, customerName, pizzaId, pizzaName, orderTime, quantity, slices) which has the following FDs:
   customerId → customerName
   pizzaId → pizzaName
   customerId orderTime pizzaId → quantity slices
   Project these FDs onto the relation X(customerId, customerName, quantity, orderTime).
   In other words, suppose relation $X = \pi_{customerId,\ customerName,\ quantity,\ orderTime}(PizzaOrders)$. *Determine a minimal basis of FDs that hold in* X.

   (b) Suppose we have a relation TakeCourses(studentId, studentName, profId, profName, courseNumber, department, capacity) which has the following FDs:
   studentId → studentName
   profId → profName
   courseNumber department → profId capacity
   Project these FDs onto the relation Y(profName, studentName, courseNumber, department).

   (c) Suppose we have a relation $R(A, B, C, D, E, F)$ which has the following FDs:
   $B \to A$, $E \to C$, $F \to D$, $CD \to B$, $CF \to A$.
   Project these FDs onto the relation $Z(A, E, F)$.

3. Consider the relation $R(A, B, C, D, E)$ with FDs $C \to AD$, $DE \to B$, $A \to E$, and $B \to C$.

   (a) List all the keys of R.

   (b) Indicate which FDs are BCNF violations.

(c) Decompose R, if necessary, into relations that are in BCNF. You can choose any BCNF violations you want to do the decomposition. Show your work.

(d) Is the result of your decomposition dependency-preserving? In other words, is the set of FDs that hold in the individual relations in your decomposition equivalent to the original set of FDs in R? Explain your answer. If your decomposition is not dependency-preserving, re-decompose R into 3NF. Show your work.

4. Consider a relation Stocks(B, O, I, S, Q, D), whose attributes may be thought of informally as broker, office (of the broker), investor, stock, quantity (of the stock owned by the investor), and dividend (of the stock). Let the set of FDs for Stocks be $S \rightarrow D$, $I \rightarrow B$, $IS \rightarrow Q$, and $B \rightarrow O$.

   (a) List all the keys for Stocks.

   (b) Use the 3NF synthesis algorithm to find a lossless-join, dependency-preserving decomposition of Stocks into a set of 3NF relations. Show your work.

   (c) Are any of the relations in your final decomposition not in BCNF? If yes, decompose them into BCNF.

5. Consider the following relation (similar to the movie database in the textbook):

   StarringMovie(actorName, actorAddress, studioName, studioAddress, title, year, budget, role, wage)

   Consider the following constraints for this relation:

   Constraint 1: Every actor has a unique name and address.

   Constraint 2: Every studio has a unique name and address.

   Constraint 3: Every movie has a release year, budget and a unique movie name, and it is owned by only one studio.

   Constraint 4: An actor can play multiple *different* roles in the same movie. For example, an actor can play the roles of "Teacher" and "Policeman," but cannot play two "Students" in a movie. Multiple actors can play the same role in the same movie (so it is possible that there may be many actors playing the role of "Student" in a movie.).

   Constraint 5: If multiple actors play the same role in the same movie, they must all get the same wage. For example, all "Students" in the same movie should get the same daily wage. Note that if someone plays a "Student" in two different movies, they may get different daily wages for each of those jobs. Also, if different actors play different roles in the same movie, they may get different wages, e.g., someone playing a "Teacher" may get more or less money than someone playing a "Student" in the same movie. Someone playing multiple roles in the same movie will have multiple wages listed, one for each role.

   (a) Based on the statements, list all the non-trivial functional dependencies for this schema. For each FD you write, list the statement number(s) from which you derived that FD. A minimal basis of FDs is fine.

   (b) Is StarringMovie in BCNF? If not, convert it to BCNF. Show your work.