Virginia Tech. **CS 4604 – Introduction to DBMS**
Computer Science **Spring 2013, Prakash**

Homework 4: FDs and NFs

(due April 10th, 2013, 9:05am, in class—hard-copy please)

Reminders:

a. Out of 100 points.
b. Rough time-estimates: ~3-5 hours.
c. Please type your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.
d. There could be more than one correct answer. We shall accept them all.
e. Whenever you are making an assumption, please state it clearly.

Q1: Inferring FDs [27 points]

Consider the following schemas and sets of functional dependencies that hold in those schemas. It is enough for you to list only completely non-trivial FDs with a single attribute on the right hand side.

1). R1(A, B, C, D) with FD's C → B, C → A and A B → D.

Q1.1.1 (3 points) What are all the nontrivial FDs that follow from the given FDs?

Q1.1.2 (3 points) What are all the keys of R1?

Q1.1.3 (3 points) What are all the superkeys for R1 that are not keys?

2). R2(A, B, C, D) with FD's A B → C, C → D and D → B.

Q1.2.1 (3 points) What are all the nontrivial FDs that follow from the given FDs?

Q1.2.2 (3 point) What are all the keys of R2?

Q1.2.3 (3 points) What are all the superkeys for R2 that are not keys?

3). R3(A, B, C, D, E, F) with FD's E → F, E → D, A D → B, A D → C, and F → B.

Q1.3.1 (3 points) What are all the nontrivial FDs that follow from the given FDs?

Q1.3.2 (3 points) What are all the keys of R3?

Q1.3.3 (3 points) What are all the superkeys for R3 that are not keys?

Q2: Projection of FDs [12 points]

For all the parts of this question, again you only need to write down FDs which are completely non-trivial and which have single attributes on the right hand side. Also, in your result, a minimal basis is enough.

Q2.1 (4 points) Suppose we have a relation OrderPizza(customer_id, customer_name, pizza_id, pizza_name, ordertime, quantity, slices) which has the following set of FDs:

$$\text{customer\_id} \rightarrow \text{customer\_name},$$
$$\text{pizza\_id} \rightarrow \text{pizza\_name},$$
$$\text{customer\_id pizza\_id ordertime} \rightarrow \text{quantity slices}$$

Project these FDs onto the relation X(customer_id, customer_name, quantity, ordertime).

Q2.2 (4 points) Suppose we have a relation TakeCourses(student_id, student_name, professor_id, professor_name, course_number, department, capacity) which has the following set of FDs:

$$\text{student\_id} \rightarrow \text{student\_name},$$
$$\text{professor\_id} \rightarrow \text{professor\_name},$$
$$\text{course\_number department} \rightarrow \text{professor\_id capacity}$$

Project these FDs onto the relation Y(professor_name, student_name, course_number, department).

Q2.3 (4 points) Suppose we have a relation R(A, B, C, D, E, F) with the following set of FDs:

$$B \rightarrow A,\ E \rightarrow C,\ F \rightarrow D,\ C\ D \rightarrow B \text{ and } C\ F \rightarrow A$$

Project these FDs onto the relation Z(A, E, F).

Q3: 3NF Decomposition [10 points]

Consider a relation **Stocks**(B, O, I, S, Q, D), whose attributes may be thought of informally as broker, officer (of the broker), investor, stock, quantity (of the stock owned by the investor), and dividend (of the stock). Let the set of FD's for **Stocks** be:

$$S \rightarrow D,\ I \rightarrow B,\ IS \rightarrow Q, \text{ and } B \rightarrow O$$

Q3.1 (3 points) List all the keys for **Stocks**.

Q3.2 (5 points) Use the 3NF synthesis algorithm to find a lossless-join, dependency-preserving decomposition of **Stocks** into a set of 3NF relations. Please show your steps.

Q3.3 (2 points) Are any of the relations (that you get in Q3.2 above) not in BCNF?

Q4: FDs in English Language [20 points]

Consider the following relational schema (similar to the MOVIE database in the textbook):

StarringMovie(actor_name, actor_address, studio_name, studio_address, title, year, budget, role, dailywage)

Consider also the following constraints in English:

Statement 1 Every actor has a unique name and an address.

Statement 2 Every studio has a unique name and an address.

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

Statement 4 An actor can play multiple **different** roles in the same movie. For example, an actor can play a "teacher" and a "policeman", but cannot play two "students" in a campus/school 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 "students" in a movie.).

Statement 5 If multiple actors play the same role in the same movie, they **must** get the same daily wage. For example, all "students" in the same movie should get the same daily wage. Note that if they play a "student" in different movies, they may get different daily wages. Also, if different actors play different roles in the same movie, they **may** get the different daily wages, e.g., someone playing a "teacher" may get more or less per day than someone playing a "student" in the same movie.

Q4.1 (2 points) Is this a good relational design or not? Explain your answer: without going into any of the normal forms, list the different types of anomalies this relation has.

Q4.2 (8 points) Based on the English description, list all the non-trivial functional dependencies for this schema and list also the corresponding statements you

used to formulate each FD. (**Note**: a minimal basis is enough and some FDs may come from multiple statements).

Q4.3 (10 points) Is the above schema in BCNF? If not, try to convert it into BCNF first. Is your result of decomposition dependency-preserving? If it is not dependency preserving, explain your answer, and then transform it into 3NF. Show your steps.

Q5: Normal Forms and Implied FDs [31 points]

Consider the schema relation R(A, B, C, D, E) and the following set of FDs:

$$C \rightarrow A\ D,\ D\ E \rightarrow B,\ A \rightarrow E \text{ and } B \rightarrow C$$

Q5.1 (4 points) List all the keys of R.

Q5.2 (2 points) Indicate all the BCNF violations, if any.

Q5.3 (5 points) Decompose the relations, as necessary, into collections of relations that are in BCNF. You can choose any of BCNF violations that you listed in Q5.2 above to decompose. Please show your steps.

Q5.4 (12 points) For any given relation X with set of FDs F holding in it, are we required to consider FDs that are not in F but follow from it (i.e. the 'implied FDs') to figure out if the relation violates BCNF? If YES, give an example. If NO, prove why we don't need to.

Hint: If an implied FD $Y \rightarrow Z$ violates BCNF, can we deduce anything about any of the FDs in the given set F?

Q5.5 (8 points) Is the result of your decomposition (in Q5.3) dependency preserving? Explain your answer. If it is not dependency preserving, then transform it into 3NF. Make sure you show your steps.