Fall 2010 CS 3200 Class Project: Milestone 4

The goal of this milestone is to practice schema refinement and check our project schema to see if all relations are in some desirable normal form.

This milestone is to be completed individually (i.e., no teams). You can discuss problems with other students, but you have to create all deliverables yourself from scratch. In particular, it is not allowed to copy somebody else's code or text and modify it.

The report for this milestone is due on Friday, **October 15 at 5pm**. For late submissions you will lose one percentage point per hour after the deadline. This milestone is worth 10% of your overall homework score. Please email the deliverables to both me and Yue. You should receive a confirmation email from either of us. If you have not received a confirmation email within 12 hours after submitting your solution or by the time of the deadline, whichever comes first, you need to email us immediately to make sure we actually received your submission. (Of course, if you submit too close to the deadline, you might receive a confirmation sometime within the next 30-60 minutes after you submitted.) If you need to send multiple files, please create a single zip file. Many other attachments types, in particular rar files, are rejected by the CCIS mail server.

Book Exercises

1. Solve exercise 19.2. For part 1, show the steps of your computation for finding all keys. For parts 2 and 3, justify your answer.

Hints for part 1: A key determines all attributes of a relation, i.e., its *attribute closure* is identical to the entire relation schema. Hence, to find all superkeys you can use the algorithm for computing the attribute closure and run it for each possible set of attributes. (See page 614.) However, notice that we are looking for keys, not superkeys. A key is a minimal superkey, e.g., once you know that AB is a key, all supersets like ABC, ABD, ABCD are not keys, because they are not minimal. If you start searching for the smallest keys first, you might be able to save yourself a lot of computation. E.g., after you discovered that AB is a key, you do not need to run the attribute closure algorithm for any of its supersets any more.

Hints for part 2, 3: Testing if a schema is in 3NF or BCNF in general can be very expensive. If you suspect that the schema is not in a certain normal form, you can show this by finding an FD that violates that normal form. It is always easiest to start with those FDs given, instead of deriving the closure immediately. (However, always consider the standardized form of the given FDs, where you have only a single attribute on the right-hand side. This is step 1 of the algorithm on page 626.) Showing that a schema actually is in 3NF or BCNF is harder. For 3NF, one can often base an argument on the third condition (for X→A, attribute A is part of some key). E.g., considering the given set of FDs, there are sometimes only few options for the (single!) attribute A of the right side of an FD in the closure.

2. Exercise 19.6.

3. Exercise 19.10.

Analyzing the Project Schema

4. List all FDs that hold for the schema given by the Milestone 3 reference solution (mandatory part of Milestone 3 only, not the alternative designs), including primary key constraints. You can identify FDs from the application description given in Milestone 1, and based on common sense.

Based on the FDs you found, identify any relations that are not in BCNF and perform a lossless-join decomposition to create smaller relations that are each in BCNF.

Deliverables

Create a report with the solutions to the four problems.