CS 360: Database Management Systems

Department of Computer Science
University of Idaho

Hasan M. Jamil | **Tutorial Questions 2c** | Semester: Open

1. Consider a relation scheme $R(A, B, C, D)$. Assume that $A \rightarrow CD$ and $B \rightarrow D$ holds on every instance r of R. Show by a single counterexample (a relation instance with at most 4 tuples) that $B \rightarrow C$ and $C \rightarrow B$ are not logically implied by $A \rightarrow CD$ and $B \rightarrow D$.

   (Note: You will receive extra credits for solving it with only two tuples.)

2. Prove or disprove the following inference rules for functional dependencies. A proof can be either by a proof argument or by using the inference rules we have introduced in the class. A disproof should be done by demonstrating a relation instance that satisfies the functional dependencies in left-hand side of the inference rule but does not satisfy the dependencies in the right-hand side. In the following, X, Y, Z, W denote sets of attributes.

   (a) $\{W \rightarrow Y, X \rightarrow Z\} \models \{WX \rightarrow Y\}$
   (b) $\{X \rightarrow Y, X \rightarrow W, WY \rightarrow Z\} \models \{X \rightarrow Z\}$
   (c) $\{XY \rightarrow Z, Y \rightarrow W \models \{XW \rightarrow Z\}$
   (d) $\{X \rightarrow Z, Y \rightarrow Z\} \models \{X \rightarrow Y\}$
   (e) $\{X \rightarrow Y, XY \rightarrow Z\} \models \{X \rightarrow Z\}$

3. Consider a relation with scheme $R(A, B, C, D, E, H)$ and set of functional dependencies $F = \{A \rightarrow B,\ BD \rightarrow H,\ E \rightarrow D,\ C \rightarrow AE\}$.

   - Prove using inference rules that $F \models ADC \rightarrow H$.
   - Compute $\{BE\}^+$.
   - Compute all candidate keys of R.

4. Compute the canonical cover of the set $F = \{A \rightarrow BC, A \rightarrow E, ACD \rightarrow E, B \rightarrow D, E \rightarrow A\}$ of functional dependencies over the relation scheme $R = \{A, B, C, D, E\}$.

5. Given $\mathbf{R} = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, A_{11}\}$ with the FDs $\{A_9 \rightarrow A_{11}, A_1A_9 \rightarrow A_2A_6A_7,\ A_9A_3 \rightarrow A_1A_4A_5, A_2A_9A_7 \rightarrow A_3A_{10}, A_{11} \rightarrow A_8A_1\}$, find a canonical cover of this set of FDs. Find a dependency-preserving and lossless join 3NF decomposition of $\mathbf{R}$. Is there a BCNF decomposition of $\mathbf{R}$ that is both dependency-preserving and also loseless? If so, find one such decomposition. If not, justify your answer.

6. Suppose we have the following requirements for a university database that is used to keep track of students' transcripts:

   (a) The university keeps track of each student's name (SNAME), student number (SNUM), social insurance number (SIN), current address (SCADDR), permanent address (SPADDR), birth date (BDATE), sex (SEX), year (YEAR) (freshman, sophomore, ..., graduate), major department (MAJORDEPTCODE), minor department (MINORDEPTCODE) (if any), and degree program (PROG) (B.A., B.S., ..., Ph.D.). Both social insurance number and student number have unique values for each student.

   (b) Each department is described by a name (DEPTNAME), department code (DEPTCODE), office number (DEPTOFFICE), and office phone (DEPTPHONE). Both name and code have unique values for each department.

   (c) Each course has a course name (CNAME), description (CDESC), course number (CNUM), number of credits (CREDIT), and offering department (CDEPT). The value of course number is unique for each course.

   (d) Each section has an instructor (PROFNAME), semester (SEMESTER), academic year (AYEAR), course (COURSE), and section number (SECNUM). Section number distinguishes different sections of the same course that are taught during the same semester/year; its value are 1, 2, ..., up to the number of sections taught during each semester.

   (e) A transcript refers to a student (SSIN), a particular course, a particular section, and a grade (GRADE).

   Design a relational database schema for this database application. First show all the functional dependencies that should hold among attributes. Then design relation schemas for the database that are each in 3NF or BCNF. Specify the primary key of each relation. Note any unspecified requirements, and make appropriate assumptions to make the specification complete. (Remember that, all specifications are usually incomplete and should not be surprising.)

   Note: More credits for BCNF.