# Harvey Mudd College
# Computer Science 133
# Database and Knowledgebase Systems
# Spring Semester 2000

## Assignment #4 – Relational Design
Due 5:00pm, Friday March 3, 2000

## Part I - Functional Dependencies

1. List all of the non-trivial functional dependencies that are satisfied by the following relation instance:

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_2$ | $b_1$ | $c_3$ |

2. Prove or disprove the following inference rules for functional dependencies. The proof should be by using Armstrong's Axioms (relexivity, augmentation and transitivity) and the three rules that we showed follow directly from them (union, pseudotransitivity, and decomposition). A disproof consists of presenting a relation instance that satisfies the conditions and functional dependencies in the left-hand side of the inference rule, but does not satisfy the conditions or dependencies in the right-hand side.

   (a) $\{W \to Y,\ X \to Z\} \models \{WX \to Y\}$

   (b) $\{X \to Y\}$ and $Z \subseteq Y \models \{X \to Z\}$

   (c) $\{X \to Y,\ X \to W\ WY \to Z\} \models \{X \to Z\}$

   (d) $\{XY \to Z,\ Y \to W\} \models \{XW \to Z\}$

   (e) $\{X \to Z,\ Y \to Z\} \models \{X \to Y\}$

   (f) $\{X \to Y,\ XY \to Z\} \models \{X \to Z\}$

3. Consider the following two sets of functional dependencies:

$$F = \{A \to C,\ AC \to D,\ E \to AD,\ E \to H\}$$
$$G = \{A \to CD,\ E \to AH\}.$$

   Prove that they are equivalent.

4. Consider a database scheme $\mathcal{R} = \{A, B, C, D, E\}$ in which the following functional dependencies are defined to hold: $F = \{A \to BC,\ CD \to E,\ B \to E,\ E \to A\}$.

   (a) List the candidate keys for $R$, given the constraints specified in $F$.

   (b) Place $F$ in an equivalent form $F'$ with each right hand side consisting of a single attribute.

   (c) Is $F'$ canonical? If so, state why; if not, find an equivalent canonical set of dependencies.

   (d) Suppose that we decompose the scheme $\mathcal{R}$ into $\mathcal{R}_1 = \{A, B, C\}$, $\mathcal{R}_2 = \{A, D, E\}$. Show that this is a lossless join decomposition using either the theorem or the tableau method.

   (e) Show the projection of $F'$ for each of $\mathcal{R}_1$, $\mathcal{R}_2$. Does this decomposition preserve dependencies?

## Part II - Normal Forms

Consider the database schema:

**Registrar**(id#, name, address, city, state, zip, course#, description, year, term, grade)

with the following set, $F$, of functional dependencies:

id# $\to$ name, address, city, state, zip
zip $\to$ city, state
*street*, city, state $\to$ zip
id#, course# $\to$ year, term, grade
course# $\to$ description
description $\to$ course#

where *street* is part of the address.

1. What are the candidate keys for this schema?

2. Provide a decomposition of this relation into a database schema that is in first normal form, but not second normal form. For each relation schema $R_i$, in the decomposition, show the schema, the (interesting part of) the projection of $F$ over $R_i$, the candidate keys for the schema, and whether that single schema is in 2NF or not (recall that only one relation schema need not be in 2NF for the database schema to not be in 2NF). If it is not, explain why it is not. (Note, here and in the next two problems, your solution decomposition must be lossles and dependency preserving)

3. Provide a decomposition of your solution to the last problem into a database schema that is in second normal form, but not in third normal form. For each relation schema $R_i$, in the decomposition, show the schema, the (interesting part of) the projection of $F$ over $R_i$, the candidate keys for the schema, and whether that single schema is in 3NF or not. If it is not, explain why it is not.

4. Provide a decomposition of your solution to the last problem into a database schema that is in third normal form. For each relation schema $R_i$, in the decomposition, show the schema, the (interesting part of) the projection of $F$ over $R_i$, and the candidate keys for the schema.

5. Is your solution to the last problem in Boyce-Codd Normal Form. Why or Why not? If not, is there a lossles, dependency-preserving decomposition into BCNF?