Introduction to Database Design IT University, Spring 2012

Week 8

Exercises

These exercises are not mandatory.

Exercise 1: BCNF

Consider a database consisting of a single table

screening(theater_id, theater_name, movie_id, movie_title, start_time, date, screen) .

A tuple in this table represents a screening of a movie, for example, the tuple

(234, 'Palads', 7563, 'Hamilton', '19:00', '2012:03:19', 7) .

represents a screening of the movie 'Hamilton' in Palads screen 7, on March 19 at 19:00.

Write down some functional dependencies that you expect to hold for the table *screening*. First, try to come up with as many functional dependencies as you can think of, then try to give a minimal set that describes the situation. Give a candidate key for the table *screening*, and give an example of a superkey that is not a candidate key.

Is the table *screening* on BCNF? Try to give a formal argument. If it is not on BCNF, then suggest a better design, and argue that it is BCNF.

Exercise 2: Understanding 3NF

Big Kahuna Burger is a chain of burger restaurants consisting of more than 10000 branches (or locations) worldwide. It is company policy that the manager of each location selects an employee of the month every month, since studies show that this increases morale among employees. The chain plans to keeps a central database containing information on employees including who is employee of the month.

Jules Winnfield has been employed to design the database. He considers two possible designs.

- Design A: *employee(id, name, salary, branch_id)*, *employee_of_the_month(year, month, employee_id, branch_id)*
- Design B: *employee(id, name, salary, branch_id)*, *employee_of_the_month(year, month, employee_id)*

In both designs *id* is the key of the *employee* table (in particular, an employee can only work for one branch). There are two important integrity constraints for the database: there can not be two different employees of the month in the same branch for the same month, and an employee can only be employee of the month at the branch where (s)he is employed.

- Express the first of these integrity constraints as a functional dependency.
- Give two different candidate keys for table *employee_of_the_month* of design A, give one for design B.
- Are the designs in BCNF? Are they in 3NF? Argue for your answers.

Hint. When showing that a table is 3NF it suffices to show that all attributes are contained in candidate keys.

Suppose that we want to insert the information that employee number 34856 is employee of the month for March 2011 at branch number 2387. Assuming that the integrity constraints are upheld by the database before the insertion, what do we need to check if we want to make sure that the integrity constraints are not violated by the insertion? **Write queries** that obtain the information you need information to determine if the insertion is safe. Answer the question for both designs.

Now, the punchline of this exercise is that queries involving joins are expensive and we would like to avoid having to make such queries before each insertion.

Exercise 3: Designing databases

Do exercise 7.22 in DSC. I repeat it here, since some of you may have older versions of the book, where exercise numberings are different.

Design a database for a world-wide package delivery company (e.g., DHL or FedEX). The database must be able to keep track of customers (who ship items) and customers (who receive items); some customers may do both. Each package must be identifiable and trackable, so the database must be able to store the location of the package and its history of locations. Locations include trucks, planes, airports, and warehouses.

Your design should include an E-R diagram, a set of relational schemas, and a list of constraints, including primary-key and foreign-key constraints.

Note. A relational schema is what I sometimes call a database design or database schema.