NYU Tandon School of Engineering
Computer Science and Engineering
CS6083, Fall 2018

October 29, 2018

Problem Set #3 (due 11/13)

Problem 1:

Consider the bakery database that was already used in class, given by the following database schema:

Customer (custid, firstname, lastname, ccn, cphoneno, address, city, zip);
Cake (cakeid, cakename, slices, status, price);
Ingredient (ingredid, iname, price, available);
Contain (cakeid, ingredid, qty);
Orders (custid, cakeid, ordertime, pickuptime, pricepaid);

The bakery offers different types of cakes. A cake has a name, current price, and number of slices (servings) per cake. Customers need to make orders one day or more in advance, and then pick them up on the agreed day. Each cake contains a number of ingredients (e.g., flour, sugar, eggs) that are needed to make it. The *Contain* table states how much of each ingredient is needed for each cake, while the *Ingredient* table stores the cost per pound, and the number of pounds available, of each ingredient. Finally, table *Cake* has a field *status* that can be set to either *available* or *discontinued*, depending on whether a cake is still being offered, or has been discontinued. The *ordertime* and *pickuptime* attributes should store both time and date information.

In this problem, you have to design a simple web frontend with PHP, which allows people to access some of the information by using a web browser. Your web application should support the following:

(i) On a start page, there is a text box where users can type their customer id, and a button.

(ii) When the button is pressed, your PHP code should display a list of all orders by this customer, with time and date of the order, cakename, and the price that was paid.

(iii) On this page, the user should be able to click on a cakename, and get a list of all ingredients that are in this cake, and the amount.

Test your application using data provided on the course page to populate your database. You will have to meet with a grader to give a quick demo. (Details on scheduling demos will be announced later, and demos will be a few days after the deadline for the other problems.)

Problem 2:

Consider a relational schema R = (A, B, C, D, E, H, I) satisfying the functional dependencies F = {A -> BCE, C -> DB, H -> EIA, D->CH, A -> CA}.

a) Derive all candidate keys for this schema.
b) Derive a canonical cover of the functional dependencies in F
c) Is the above schema in BCNF? Prove or disprove. If it is not in BCNF, convert it into BCNF.
d) Is the BCNF schema from c) dependency-preserving? Prove or disprove. If not, convert it into 3NF.

Problem 3:

Consider the following single-table database schema for the actor-movie database, as already given in the slides for Chapter 8:

ActorMovie (aid, aname, mid, mtitle, rolename, hours, payph)

Assume there is a unique aid per actor, and a unique mid per movie. Also, an actor can play several roles in the same movie, these roles must have different role names. Also, different actors playing the same role (role name) in the same movie must be paid the same per hour (payph).

a) Explain why the above is not a good relational design. Name several reasons.
b) Identify the set F of non-trivial functional dependencies for this schema. (It is enough to identify a subset E such that the closures of E and F are the same.)
c) Derive all candidate keys for this table.
d) Derive a canonical cover of the functional dependencies in F.
e) Is the above schema in BCNF? Prove or disprove. If it is not in BCNF, convert it into BCNF.
f) Is the BCNF schema from e) dependency-preserving? Prove or disprove. If not, convert it into 3NF.
g) Suppose we add an additional constraint that actors with same name must get the same pay per hour in the same movie. How would this change your answers for parts b) through f)?

Problem 4:

In this problem, you are asked to explore the metadata querying facilities in your database system. The answers may depend on which system you have installed, so make sure to state which system you are using! Try to write the following queries using the bakery database from Problem 1. Please submit screenshots of the result.

a) For each data type that is used in the schema, list the number of attributes that have this data type.
b) List all pairs of tables that have at least one attribute in common.
c) List the number of views defined in this schema.
d) List any customers whose last name is equal to the name of a table. (For example, imagine there could be a customer called “Chris Cake” or “Igor Ingredient”.)
e) For each table, output the attributes contained in its primary key.
f) Output all attributes whose name contains the substring “name”.