NYU Tandon School of Engineering **October 31, 2017**
Computer Science and Engineering
CS6083, Fall 2017

Problem Set #3 (due November 13, except for Problem #1)

Problem 1:

Consider a database modeling an online restaurant booking system, where a customer can search restaurants and book them. It is given by the following highly simplified database schema:

CUSTOMER(cid, cname, phone)
RESTAURANT(rid, rname, raddress, description, capacity)
BOOKING(bid, cid, rid, btime, quantity)

Each customer has a unique cid, along with a cname and a phone number. Each restaurant has a unique rid, along with a rname, raddress, and a short description such as "Chinese Food" or "Chicken, Sandwiches, Wings". The capacity indicates how many seats this restaurant. (We assume that parties can share tables as needed, or seats can be moved between tables, so we only keep track of seats, not tables.) For the booking table, it has a unique bid, and btime is the start time for the booking. You should assume that every booking starts on the hour and lasts for one hour (e.g., 2-3pm, 3-4pm, etc.). The quantity is the number of seats requested.

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

(i) On the start page, there is a box where users can type a customer name, a keyword, and the number of people. Also the customer can enter or select the date and time they want to book for.

(ii) After the button is pressed, your PHP code should provide a list of all **available** restaurants which have the keyword in their descriptions or names, along with detailed information of each restaurant. If no keyword is entered, all available restaurants should be displayed.
Note: To simplify this problem, we assume every reservation is for one hour. Thus, to check if a restaurant is available, you can simply use the capacity minus the sum of the quantity already booked within that hour.

(iii) Next, the user should be able to click a button to book a table. After that, your code should add a record into the booking table and turn to another page which shows all previous bookings of this customer.

Test your application, using the data provided on the course page to populate your database. You will meet with the 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, F, G) satisfying the functional dependencies
F = {BF -> C, CF -> D, G -> A, G->E, FG -> AD}

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. In 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 modeling a patient-treatment database at a hospital:

PatientTreatment (pid, pname, page, psate, docid, docname, docage, doclevel, docsalary, did, dname, dtype, description, rid, rname, rcapacity, ttime, tcost, insurancediscount)

So this table has information about a patient (pid, pname, page, pstate), a doctor (docid, docname, docage, doclevel, docsalary) treating that patient, a disease (did, dname, dtype, description), and the room where the treatment (or procedure) is performed (rid, rname, rcapacity). Thus, each record contains all the information for one procedure (say, doing an examination, giving an injection or performing a minor surgery) for one patient, where we also store the ttime, tcost and the insurance discount of the treatment that was done. (The insurance discount is the amount paid by the insurance, so the customer has to pay the rest.) In addition, we assume that the following conditions hold:

(i) For each room, it can be used for only one particular type of disease.
(ii) All doctors at the same level (e.g., resident, junior, senior) must have the same salary.

a) Explain why the above is not a good relational design. Name several reasons!
b) Derive all candidate keys for this table.
c) 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.)
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 for any two patients who live in the same state and spend the same tcost for a treatment (or procedure), the insurance discount should also be the same. How would this change your answers to parts (b) to (f)?

Problem 4:

In this problem, you are asked to explore the metadata querying facilities in your database system. Thus, the answers may depend on which system you have installed. So remember to state which system you are using! Try to write the following queries using the restaurant booking database from Problem 1:
a. List all tables in the schema, and the attribute names of their primary keys.
b. List the table with the most attributes.
c. List the attribute name and table name of any attribute of type int.
d. List all pairs of tables that have an attribute with the same name and same data type.