NYU Tandon School of Engineering
Computer Science and Engineering
CS6083-INET, Spring 2018

March 24, 2018

Homework #3 (due April 6)

Problem 1 (15 points):

In this problem, you are asked to build a small web interface for a database using PHP. For testing, you can use the Student-University Database schema and data available at NYU Classes in the resources section (though most other schemes also work). In particular, the goal is to use meta data features in order to allow the user to explore the structure of the tables, rather than just the data in it. In particular, you need to implement the following operations:

a) A user should be able to input a string into a text field in a web page, and the system should return the name of any table that contains this string in its name, using the LIKE operator. Thus, using the Student-University schema, if the user types in "stu" it should return "Student" and "Student_University" (you may ignore upper versus lower case).

b) When the user clicks on the name of a table displayed as a result of a) above or c) below, the system should display the names of all attributes in that table.

c) When the user clicks on an attribute name displayed in b), the system should display the names of all tables containing this attribute.

You can review the online tutorial at http://www.w3schools.com/php/default.asp for more information about PHP and for writing interactive sample scripts. Submit one paragraph describing what you did as part of the homework. There may also be a short demo, which the TA will conduct during his office hours.

Problem 2 (15 Points):

Consider the following database schema:

Customer (cid, cname, cphone, ccity);
Landlord (lid, lname, lphone, lcity);
Residence (rid, rname, rstate, rcity, raddr, rtype, rarea, lid, status); //Status: Available, Rented
Leases (cid, rid, startdate, enddate, price);
LeasePayment (cid, rid, paymentdate, price, penalty);
Rating (cid, rid, rtime, score);

a. Create a view to show for each landlord the total amount of penalty earned for each year.
b. Create a trigger that will make customers pay a penalty of $5/day if the lease payment is done after 5th of every month.
c. Create a trigger to change the status of Residence from Available to Rented and vice versa once the start/end date has passed.

Problem 3 (20 points):

Here is the table containing the information for treatment of each patient in a hospital:

Patient_Disease_Treatment (pid, pname, page, pstate, docid, docname, docage, doclevel, docsalary, did, dname, dtype, description, rid, rname, rcapacity, admittime, daysadmitted, tcost, discount, patienttype)

This table has information related to each patient, doctors and disease the patient was suffering from. Additionally, it has the information related to the room the patient was admitted to. The treatment related information like cost, discount given to a specific type of patient, in case if it's a senior citizen or a pregnant woman, the amount of time the patient was admitted all is included here. For simplicity, we can assume that the doctors at same level have same salary.

(a) Explain why this is not a good relational design.

(b) Identify all non-trivial functional dependencies F for this schema.

(c) Derive a canonical cover of the functional dependencies in F.

(d) Is the above schema in BCNF? Prove or disprove. If it is not in BCNF, try to convert it into BCNF.