COS 597D, Fall 2013 – Assignment 3

Due at 1:30pm Wednesday, October 2, 2013

Collaboration Policy

You may discuss problems with other students in the class. However, each student must write up his or her own solution to each problem independently. That is, while you may formulate the solutions to problems in collaboration with classmates, you must be able to articulate the solutions on your own.

Late Penalties

- 10% of the earned score if submitted after class but by 11:59 pm Wed.
- 20% of the earned score if submitted by 11:59 pm on Fri. 10/4/13.
- 40% of the earned score if submitted after 11:59 pm on Fri. 10/4/13.

Problem 1.
Part a: Let R be a relation with attributes (a,b,c,d) over domains A, B, C, and D, respectively. Let {a} and {b,c} be two candidate keys for R. Let {b} be a foreign key referencing attribute x, the primary key of relation X. Let Q be a relation with attributes (e,f,g,h) over domains A, B, C, and D, respectively. Let {g,h} be a candidate key for Q, and let {e,h} be a foreign key referencing attributes {y,z}, the primary key of relation Y. What candidate key and foreign key constraints ***must*** be true of R-Q?

Part b: Let R again be a relation with attributes (a,b,c,d) over domains A,B,C, and D, respectively. Again let {a} and {b,c} be two candidate keys for R, and let {b} be a foreign key referencing attribute x, the primary key of relation X. Let T be a relation with attributes (c,d) over domains C and D, respectively. Let {c} be a candidate key for T and let {d} be a foreign key referencing attribute u, the primary key of relation U. What candidate key and foreign key constraints ***must*** be true of R÷T?

Problem 2.
For this problem we will use the following relational database (based on one in *Database System Concepts* by Silberschatz, Korth and Sudarshan).

- relation ***company*** with attributes
  (co_name, headquarters_street, headquarters_city, URL)
- relation ***branch*** with attributes
  (*co_name, branch_street, branch_city, manager)*
  - *co_name* is a foreign key referencing ***company***

We are interested in the query

Find the (co_name,URL) pairs of all companies with a branch located in every city in which "Walmart" has a branch.

Part a: Express the query in the relational algebra. You may use derived operations such as natural join.

Part b: Express the query in the tuple relational calculus.

Problem 3
In slide # 13 of the SQL presentation (posted under 9/25/13), we considered the SQL query "*Find names of all branches with accts of cust. who live in Rome"*:

```
SELECT A.bname
FROM Acct A
WHERE A.acctn IN (SELECT D.acctn
                  FROM Owner D, Cust C
                  WHERE D.name = C.name AND C.city='Rome')
```

Call the result of this query ***Result_in_Rome***.

Write an SQL query that does ***not*** use ***nested*** SELECT for ***Result_in_Rome***.