COS 597A, Fall 2011 -Problem Set 2

Due at 3:00pm Wednesday, October 12, 2011

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

- 5% of the earned score if submitted after class but by 5:45pm the day due.
- 20% of the earned score if submitted by 5pm on Friday 10/14/11.
- 40% of the earned score if submitted by 5pm on Monday 10/17/11.
- **No credit** if submitted later than the 40% penalty deadline.

Chapter, exercise and page numbers refer to the course text *Database Management Systems*, 3rd edition, by Ramakrishnan and Gehrke

---

1. (10 points)
Give an example of two relations R and Q and a tuple of R that is not in $(R \div Q) \times Q$. You must specify all the tuples in R and Q. It is fine for R and Q to be small, but $R \div Q$ must not be empty.

2. (20 points)
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?

3. (30 points)
For this problem we will use the following relational database (this database and some of the questions come from the recommended text *Database System Concepts* by Silberschatz, Korth and Sudarshan).

- relation ***employee*** with attributes (*name, street, city*)
- relation ***company*** with attributes (*co_name, city*)
- relation ***works*** with attributes (*name, co_name, salary*)
    - *name* is a foreign key referencing ***employee***
- relation ***manages*** with attributes (*employee_name, manager_name*)
    - *employee_name* is a foreign key referencing ***works***
    - *manager_name* is a foreign key referencing ***works***

Express the following queries with relational algebra expressions. You may use any relational algebra operations, including intersection, join and division.

i. Find the names of all employees who work for Microsoft and draw a salary of no more than $30,000.
ii. Find the names and employers' names of all employees who live in Trenton and draw a salary of more than $1,000,000 per year.
iii. Find the names of all companies that do not have a location in Princeton.
iv. Assume companies may be located in several cities. Find the names of all companies located in every city in which Fred's Pizza Co. is located.
v. Find the names and addresses of all employees who earn more than every manager of IBM.

4. (30 points) Express the queries of Problem 3 in the tuple relational calculus.

5. (10 points) The relational algebra and tuple relational calculus are equivalent in expressive power. To prove this constructively, we need an algorithm to produce an equivalent relational algebra query from any tuple relational calculus query and an algorithm to produce an equivalent tuple relational calculus query from any relational algebra query. Parts a and b below ask you to provide small steps in constructing such algorithms.

Part a: Below is a generic tuple relational calculus query using "there exists" in a straightforward way. Give an equivalent relation algebra query. What captures the sense of "there exists" in the relational algebra query?

Let rel_1, rel_2, … rel_k be k base relations in a database. Let $attr_1$ be an attribute of rel_1, $attr_2$ be an attribute of rel_2, etc. such that the attributes have distinct names.

Generic tuple relational calculus query:

$$\{T \mid \exists R_1 \ \exists R_2 \ldots \exists R_k \ (R_1 \in rel_1 \wedge R_2 \in rel_2 \wedge \ldots \wedge R_k \in rel_k \wedge T[attr_1] = R_1[attr_1] \wedge T[attr_2] = R_2[attr_2] \wedge \ldots \wedge T[attr_k] = R_k[attr_k]) \}$$

Part b: Let rel_1 and rel_2 be two base relations in a database. Give an equivalent tuple relational calculus query for $rel_1 \times rel_2$.