Assignment 3

Due December 20, 2018

Question 1: Incomplete Databases

Question 1.1

In class, we have seen examples of SQL queries $q_1$ and $q_2$, such that $q_1$ and $q_2$ are logically equivalent if we ignore the existence of NULL, but return different results an a database $D$ with NULLs. Find a different “interesting” example of such $q_1$, $q_2$ and $D$. The example should be given as two text files, d.sql and q1q2.sql, that we should be able to paste into the left and right frames, respectively, of http://sqlfiddle.com under PostgreSQL9.3.

Question 1.2

Let $I$ be a v-instance, and let $Q(x_1,\ldots,x_k)$ be a UCQ.

Part 1. Prove that the following algorithm computes precisely the certain answers of $Q$ on $I$.

1. Evaluate $Q$ over $I$, assuming that each labeled null is a unique (fresh) constant.
2. Remove from the result of the first step every tuple that contains one or more labeled nulls.

Part 2. Would the above algorithm compute the certain answers if we allowed $Q$ to use the built-in predicate $x \neq \tau$? Here, $x$ is a variable and $\tau$ is either a variable or a constant. Please explain your answer.

Question 2: Data Exchange

Imagine a situation where there is a public relational schema $S$ with a known set $\Sigma$ of integrity constraints, consisting of functional dependencies and a weakly-acyclic collection of inclusion constraints. Now, suppose that a database $D$ over $S$ is stored in a remote server that does not provide us with full access to $D$, but rather partial views that are represented as pairs $(q_i, r_i)$, for $i = 1,\ldots,k$, where $q_i$ is a conjunctive query and $r_i$ a relation, with the guarantee that $r_i \subseteq q_i(D)$ for all $i = 1,\ldots,k$.

Devise polynomial-time algorithms for the following problems. In each problem, prove the correctness of the algorithm and explain why the running time is polynomial in the input.

1. For a UCQ $q$ over $S$, compute the answers that can be inferred from what the server has revealed. In other words, find all the tuples of $q(D)$ that would be returned no matter what the content of $D$ is, assuming that the guarantees $D \models \Sigma$ and $r_i \subseteq q_i(D)$ hold. For the complexity, assume that everything is fixed, and the input consists of only the $r_i$ (i.e., data complexity).
2. Assume that $\Sigma$ contains only functional dependencies, that each $q_i$ is acyclic, and that no relation symbol is used more than once in $q_1, \ldots, q_k$ (that is, every relation symbol occurs in at most one $q_i$ and at most once in $q_i$). Determine whether it is possible that the server has given the complete (and not only partial) views. In other words, is there any database $D$ such that $r_i = q_i(D)$ for all $i = 1, \ldots, k$? Here, assume that nothing is fixed and everything, including $S$, $\Sigma$, $q_i$ and $r_i$, is given as input (i.e., combined complexity).

Good luck!