CS 386/586 Winter 2012
Assignment 3
Due: Tuesday January 31, 2012

NOTE: As usual, unless you have some sort of emergency, this assignment MUST be turned in on time. We will discuss the answers to these queries during class on Tuesday, January 31 in order to help you prepare for Test 1.
(This assignments will not be graded before you take Test 1.)

1. Write an SQL query that lists the agents (id, first, last) who have been on a 'Secret' or 'Top Secret' mission that succeeded.

   Write this query three ways, once using EXISTS, once using IN, and once **without** using any of the special predicates in the WHERE clause that deal with subqueries (such as IN, NOT IN, EXISTS, etc.).

   Turn in all three SQL queries.

   Describe that query or queries that you issued or other steps that you performed to convince yourself that all three queries returned the exact same answer.

   Show the first five rows and the count for one of your query answers.

   Write a relational algebra query that is equivalent to any of these three SQL queries.

2. Write an SQL query that lists the country and city for which the agents from that country/city have the lowest average salary (compared to the average salary for agents from other cities and countries), along with the average salary.

   Turn in your SQL query, (at most) the first five rows), and the count of rows in the query answer.

3. Write a query that counts the number of agents that are not on a team. Your query answer should be just a count (a single value).

   Turn in your SQL query, (at most) the first five rows), and the count of rows in the query answer.

   Describe that query or queries that you issued or other steps that you performed to convince yourself that your returned the correct answer.

4. Write an SQL query that lists all of the agent_id who speak the maximum number of languages compared to the number of languages spoken by all of the agents in the Introdb_spy database. Note you must use your query to figure out the maximum number of languages spoken by all of the agents; you are NOT allowed to figure that out by hand and then use a constant in your query.)

Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

5. First, explain what the following relational algebra query computes in English. Turn in your English description.

   Then, write an SQL query that is equivalent to the following relational algebra query. (Hint: remember to use DISTINCT if your query answer has duplicate rows in SQL.)

   Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

   π agent_id (Agent – (σ city= 'Paris' Agent))

6. First, explain what the following relational algebra query computes in English. Turn in your English description.

   Then, write an SQL query that is equivalent to the following relational algebra query.

   (Hint: remember to use DISTINCT if your query answer has duplicate rows in SQL.)

   Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

   π agent_id (Agent – (σcity = 'Paris' Agent)) $\cap$ π agent_id (Agent – σ middle is NULL.Agent))

7. In relational algebra, suppose I have a table completed (s-id, c-id) which indicates that a student (identified by s-id) has completed a course (identified by c-id). Suppose I have another table Required-course (c-id) which lists the ids of the required courses. If I want to find all student ids that have completed all courses in the Required-course table, I can write in in relational algebra this way:

   Completed ÷ Required-course (or Completed / Required-course)

   I can also write it this way – without using the divide operator in relational algebra like this:

   $\pi_{\text{s-id}}$ (Completed) – $\pi_{\text{s-id}}$ (($\pi_{\text{s-id}}$ Completed) X Required-course)– Completed)

   Using this as a guide to how to compute a divide, write an SQL query for the introdb_spy

database that finds the agent_id for agents that speak all languages in the Language table. Note: there may not be any agents that speak all of those languages but you can write and execute a query that finds agents who do.

Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

8. Modify your SQL query from the previous question to find all agents that speak all of these languages: French, German, and English. Hint: for this query, you might need to use a subquery or a table-valued expression instead of the language table.

   Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

9. Write an SQL query that finds the teams where at least one of the agents speaks German.

   Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

10. Write an SQL query that finds the teams where none of the agents speak German.

    Turn in your SQL query, (at most) the first five rows of your query answer, and the total number of rows in your query answer.

    Bonus: Write an equivalent query in relational algebra.