TDBC86 - Obligatory Exercise 2
Due date: November 23, 2001 at 17.00

Given is the following relational database schema:

City(<u>name</u>, **country**, longitude, latitude, population)
Country(<u>name</u>, continent, totalPopulation, **capital**, currency)
Flight(<u>number</u>, **departure**, **arrival**)

The primary keys are <u>underlined</u>. The attribute **country** in the relation **City** is a foreign key from the **Country** relation. The attribute **capital** in the relation **Country** and the attributes **departure** and **arrival** in the **Flight** relation are foreign keys from the **City** relation. Note that cities are uniquely identified by their names[1]. Find solutions to each of the following queries in both the relational algebra and the relational tuple calculus. In your solutions you may **not** use functional operators such as count.

- 1. Find the names of countries with a total population of over 1 million.

- 2. Find the names of those cities which have flights to both 'NY' and 'London'.

- 3. Find the number, departure city, and departure country for all flights to 'LA'.

- 4. Find the flight numbers of flights departing from 'Stockholm' and arriving in a city in the 'USA'.

- 5. Find the names of countries which have at least two distinct cities with population each over 3 million.

- 6. Find the names of countries for which over 50% of the total population lives in the capital.

- 7. Find the names of cities which have flights to all of the capitals of countries in 'Europe'.

- 8. Find the names of countries that have no cities with departing flights.

- 9. Find the names of those cities in 'France' that have flights to every other city in 'France'.

- 10. Find the names of countries which have precisely two cities with populations over 3 million.

---

[1]So in an actual database 'Paris, Texas' and 'Paris, France' might be named 'Paris_T' and 'Paris_F'.

Notes:

- As stipulated in the course syllabus, this exercise may be done either individually, in a group of two, or in a group of three.

- Remember that there are point penalties for late submission. See the course syllabus.

- It is strongly recommended that you use a text-processing tool to display your results. If you write them by hand, **they must be neat**.

- It is not allowed to copy the work of others. The submission must be the original work of the individual(s) in the working group.

- The grader reserves the right to interview members of the working group about the solution.

- So that solutions may be discussed in a class meeting, students and/or groups that are very late in preparing a solution may be required to solve an alternate problem to receive credit for this exercise.

- The join notation indicated on the slides entitled "join_notation" should be used for joins. If you can not / will not use this notation then you provide a table defining your notation.

- If you have solved this problem for a previous offering of the course, you may re-use your old solution, subject to the following conditions: (a) You may not work with any partners, except possibly those with whom you worked to prepare the solution in the previous course. (b) You must explicitly note any partners from the previous course with whom you submitted a joint solution for that course. Note that grading criteria may not be identical between years, so that a solution which was found to be satisfactory last year may not be evaluated similarly this year.