5DV021 Principles of Database Systems Fall 2009
Obligatory Exercise 3
Due date: September 25, 2009 at 0800 (8am)

Given is the following relational database schema:

Airport(Code, City, Country, Latitude, Longitude)

Flight(**Airline**, FlightNumber, **Origin**, **Destination**)

Schedule(**Airline, FlightNumber**, Date, DepartureTime, ArrivalTime)

Airline(Name, Abbreviation, HomeCountry, WebSite)

Ticket(Number, **Airline, FlightNumber, Date**, Cost)

In the above schema, note the following conventions:

- The primary key of each relation is underlined.
- Additional candidate keys are wavy underlined.
- Foreign keys are shown in **boldface**. Specifically:
  - The attribute **Airline** in the Flight relation is a foreign key from the Airline relation.
  - The attributes **Origin** and **Destination** in the Flight relation are each foreign keys from the Airport relation.
  - The attribute pair (**Airline, FlightNumber**) in the Schedule relation is a foreign key from the Flight relation.
  - The attribute triple (**Airline, FlightNumber, Date**) in the Ticket relation is a foreign key from the Schedule relation.

Find solutions to each of the following queries in PostgreSQL-compatible SQL. Unless stated specifically to the contrary, the soution must eliminate all duplicates from the result. Furthermore, to keep the solutions simple and consistent, the following restrictions apply:

(i) Directives involving the keyword JOIN may not be used. Express the join conditions directly using conditions in the WHERE clause.

(ii) The solution must consist of a single SQL directive. Creation and subsequent use of temporary tables is not permitted.

(iii) Subqueries involving SELECT are allowed only in the WHERE and HAVING fields of queries and subqueries, and in the SET fields of UPDATE commands.

1. Find the code of each airport which is located either in Germany or else in Spain.

Corrected September 23, 2009

5DV021, Obligatory Exercise 3, page 2

2. Find the name and code of each airline which has both flights which depart from the airport with code 'CDG' and flights which depart from the airport with code 'ORY'.

3. Find the name and code of those airlines which do not have any flights to a city in France.

4. Find the codes of those airports which have flights to every airport in France. (Note that no French airport will normally qualify because, for example, there is no flight from 'CDG' to 'CDG'.)

5. Find the codes of those airports which have departures for exactly two distinct airlines.

6. Find the city, country and code of those airports, all of whose departures (as listed in the Flight relation) are international, in the precise sense that the destination airport is not in the same country as the airport of departure. Note that if an airport has no departures, then, trivially, all of its departures are international.

7. Find the sum of the ticket costs for each carrier for flights departing in the month of November 2009. Report 0 for those airlines with no ticket sales and order from highest to lowest. In the answer, give the name (and not the abbreviation) of the airline and the associated ticket cost.

8. Find the name of that country which has the greatest number of airports. In case of a tie, list all such countries.

Special instructions on submission:

- In addition to a hardcopy printout of the solutions, with each solution clearly marked with the query which it solves, an electronic version of the solutions containing the SQL code must be submitted, via e-mail, to `lab-5dv021@cs.umu.se`. The file must be in plain-text, orgainzed so that it is easy to extract each query and submit it to PostgreSQL. In the subject line of your mail, put OE3 and the e-mail addresses of all participants in your group. Send an actual copy of your solutions, not just a link to them. Please do not make your solutions publicly readable, as this makes it too easy for unscrupulous students to copy your work. Your work is not considered to have been submitted (for purposes of lateness) until both the paper and electronic versions are received.

- An ODBC program which genearates a test database for this exercise is provided. For each solution, also provide a printout of the result of applying your query to the test database.

Corrected September 23, 2009

5DV021, Obligatory Exercise 3, page 3

Notes:

- Remember that a correct solution must work for all instances of the database, and not just for the test database provided.
- 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 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.
- 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.
- Once a solution to this exercise is submitted, a carryover of Exercise 3 from last year is not permitted, either for points or for a satisfactory rating.