Informatics 1 Data & Analysis
Tutorial 4

Week 7, Semester 2, 2012

- Please attempt all questions on this worksheet in advance of the tutorial, and bring with you all work, including printouts of code and other results. Tutorials cannot function properly unless you do the work in advance.
- Data & Analysis tutorial exercises are not assessed, but they are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.
- Attendance at tutorials is obligatory: if you are ill or otherwise unable to attend one week then email your tutor, and if possible attend another tutorial group in the same week.
- *Related Reading:* Chapter 5 (SQL) of *Database Management Systems*, Raghu Ramakrishnan and Johannes Gehrke, 2003.

1 Introduction

In the previous tutorial you formulated some queries in relational algebra and tuple-relational calculus in the air travel domain. In this tutorial, you will formulate queries in SQL for tables of the same domain. (Please note that there are differences between the populated tables provided in the previous tutorial and this week's tutorial.) You will also interact with a database using *OpenOffice Base* using the queries you have formulated.

1.1 OpenOffice Base

In this tutorial, you will be using *OpenOffice Base.* To set up the program on a DICE machine, you need to follow these steps:

(a) Download `Comp.odb` from the course webpage:
`http://www.inf.ed.ac.uk/teaching/courses/inf1/da/`

(b) From the desktop, choose `Applications -> Office -> OpenOffice.org Base`
(alternatively, you can open a terminal window and type `oobase &`)

(c) When the *Database Wizard* window appears, choose *'Open an existing database file'*.

(d) Click `<Open>`.

(e) Locate `Comp.odb` and click `<Open>`.

The air travel data is now loaded and you are ready to start the tutorial. Make sure you regularly save any data you wish to keep.

If you wish to try the tutorial on a non-DICE machine then you may want to install your own copy of OpenOffice from `http://www.openoffice.org/`. Versions are available for Linux, Windows and Mac.

1.2 SQL Trivia

- SQL is *not* case sensitive. So keywords, table names, column names etc., can be written upper-case, lower-case or mixed upper/lower-case without affecting meaning.
- Nevertheless, often keywords are written in upper case, and it is good practice to maintain a consistent approach to the cases used in table and field names.
- In contrast to the above, strings *are* case sensitive.
- Note that strings are written using single quotation marks (apostrophes) as delimiters, *viz:* `'this is a string'`.

2 Queries in SQL

The tables in this database have the same design as the last tutorial. There are four tables — Airport, Passenger, Flight and Seat. For each of the questions do the following.

- formulate the specified query in SQL
- run the query on the Comp database
- write down the query and the number of rows in the table it produces.

To run the query on the Comp database, you need to follow the following steps:

- Click on `Queries` in the left column on the `Comp`-window.
- In the `Tasks`-view, choose *'Create Query in SQL View'*.
- Enter your query in the white space in the window that opens.
- Save the query (Ctrl+S) as `QueryN`.
- Go back to the `Comp`-window and double-click on `QueryN`. Results will be displayed in a new window.
- To edit the query, right click on it in the `Comp`-window and choose *'Edit in SQL View'*.
- When you edit a previously saved query, you will notice that Open Office Base encloses all *identifiers* (names of tables, fields and range variable) in double quotation marks (e.g., `"Airport"`, `"airportId"`). Occasionally, it seems that Open Office Base may fail to parse a query that is not written in this style, so if you encounter problems in saving or executing a query, a possible solution is to rewrite with quotation marks around identifiers. (If anyone finds an explanation for Open Office Bases' apparently temperamental behaviour in this regard, please let us know!)

Questions

(a) Retrieve all the rows in Airport table for all the airports in London city. The schema of the output table should be same as that of the Airport table.

(b) Retrieve all British and French passengers. The schema of the output table should be same as that of the Passenger table.

(c) Retrieve the names of all the passengers.

(d) Retrieve the flight number, Departure and Arrival airports of all British Airways flights.

(e) Retrieve the name of every passenger together with their flight number and the associated flight company.

(f) Retrieve all flights from all airports in London. The output schema should be same as that of the Flight table.

(g) Retrieve the ticket numbers and names of all passengers departing from London.

(h) Retrieve the flight number and flight company of all flights from London to Paris.

(i) Retrieve the ticket numbers and names of all passengers travelling in Business class.

(j) Retrieve the names and nationalities of all the Business class passengers travelling from London to Paris.

3 Tutorial Discussion

So far in this course you have used Relational Algebra, Tuple-Relational Calculus and SQL to formulate queries on tables. From your experience, is SQL more similar to Relational Algebra or Tuple-Relational Calculus? Why have a purpose-built practical query language such as SQL rather than use a theoretically clean language such as Relational Algebra or Tuple-Relational Calculus?