SQL Queries

Due date: 11:59pm Monday 2014-05-26 (beginning of Week 12)

This assignment is worth 10% of your final assessment.

This is the second stage of the major project. The aim is to set up a database using MySQL which will form the back-end of your system. We provide files to create the tables and load the data; then you need to write some SQL queries to extract appropriate information from the tables. For full marks, your answers need to be clear, and they must deal correctly with potential data sets other than the one you are given.

Academic Plagiarism declaration

By submitting this assignment you declare the following:

I declare that I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Information Technologies, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program, and that a copy of the assignment may be maintained by the service or the School of IT for the purpose of future plagiarism checking.
Retrieving your files

The files you need for this assignment can be downloaded from the website, or from the eLearning system.

Enjoy and best of luck!
To begin: Creating schema and Loading the data (No marks)

We have provided you with two MySQL scripts for creating the necessary database schema (and tables), and then loading the data files into the tables. These are exactly the same scripts we used to load the data into the Challenge site. To begin, you need to have created a user with permissions to operate on MySQL, as described in Lab08.

create_tables.sql firstly removes any existing database called assignment2 and then creates a new database in its place. It then creates four tables: one for the data about Bird species, one for data about bodymass values (measured in grams) that are reported for each kind of bird, one which gives geographic regions where the birds are distributed, and finally one which shows how often various birds have been seen on given observation dates, at a bird sanctuary, the Smithsonian Environmental Research Center, run by the Smithsonian Institute near Washington DC. The first three tables described are intended for data that has originated from the Encyclopedia of Life site eol.org (but we have cleaned it up a bit, and prepared it in format for you to bulk load as described below); the data for the fourth file was cleaned up from data obtained at serc.si.edu. Information from the two sources will be linked together using the string-valued common name of each bird.

You can run the script by logging in to MySQL, and then executing the script from within MySQL (in a command window on any Windows machines in the lab, or on your home machine if you have installed MySQL):

```
1 % mysql -u username -p
2 mysql> source create_tables.sql;
```

where username is the user you have setup (with the grant privileges statement from the MySQL lab). Of course, if you prefer, you can login to MySQL and then cut-and-paste (or re-type) the SQL commands one-by-one at the MySQL prompt.

table_load.sql uses the LOAD DATA command four times, to bulk insert the data from data files that we have given you, into the tables created above (note that this command is not part of the SQL Standard, but is supported by MySQL and some other vendors). The script and the four data files we have supplied should all be in the same directory from which you are executing your commands, and the script can then be executed from within MySQL, or its contents can be copied as SQL commands at the MySQL prompt. For example:

```
1 % mysql -u username -p
2 mysql> source load_tables.sql;
```

After loading successfully, Bird table will hold 1396 rows, Bodymass has 2756 rows, Distribution has 2264 rows, and Sighting has 802 rows.
Investigating the data (10 marks)

Now that the tables are loaded, you have to write some SQL queries to do answer some questions about the data. We suggest that you write the SQL query for each question in an appropriate .sql file (e.g. asst2.1.sql) and you can then execute each of these from inside MySQL.

1. Give the common name of any bird with a bodymass that is reported to be between 2 g and 3.3 g (inclusive) (1 mark)

2. What is the smallest bodymass recorded as being without any kind of statistic (that is, the statistic column contains NULL)? (1 mark)

3. Show the scientific name (in the form "Genus species") for any bird that is distributed in Iceland (1 mark)

4. How many sightings occurred during the 1980s of birds whose common name starts with "American "? (1 mark)

5. For every bird whose genus is Cygnus, on how many dates was it sighted? Note: it is important to include zero values for those birds in the genus that were not sighted on any date. (2 marks)

6. State the name of each genus whose members are distributed in at least 25 different locations. (2 marks)

7. Which birds were sighted without being present in the Bird table? (2 marks)

A remark on testing: The natural first step when you have written a query and (finally) removed all the syntax errors so it runs, is to check whether the rows in the output ought to be there, that is, do they have the properties that you wanted. For example, if your answer to q1 above includes the row Bee Hummingbird, then you can look in the data (either by running some simple sql queries on the database, or even by using grep etc on the input files) to see what body mass is recorded for this bird, and make sure that it is in the correct range. Of course, if a query returns many rows, you will probably only check a few of them this way. But note that this approach can check whether the output produced by a query is a subset of the correct answer, but it doesn’t give you confidence about the converse. How would you find out if your query is missing some rows it ought to generate? In particular, if a query returns no rows, maybe there really are none that satisfy the property you are looking for, but perhaps your query just had a mistake and the condition is actually true in the database – how would you know which of these is the case? The usual testing approach is to insert a few extra rows so that you know something that ought to be included in the output, and then run the query on the modified database, and check that the output has changed in the way you expected. After you do this, remember to remove the data and reload it, so subsequent tests are on the intended database.