

## COMP624121 Querying Data on the Web

### Lab Work RQ

Consider the relational version of the **Mondial** database<sup>1</sup>.

**Task 1:** Write tuple-relational calculus (TRC) expressions that, upon evaluation, return the data characterised by each of the following English-language specifications:

- (1) **Return the name of any country that has a lake.**
- (2) **Return all the information available on cities whose population is between 3 and 5 million inhabitants.**
- (3) **Return the country code and the continent of every country not in Europe or in Australia/Oceania.**
- (4) **Return the names of countries that also give their name a province.**
- (5) **Return countries that are not landlocked (i.e., have a sea coast).**

**Task 2:** Write relational-algebraic (RA) expressions that, upon evaluation, returns the data characterised by each of the following English-language specifications:

- (6) **Return countries that are not landlocked (i.e., have a sea coast).**
- (7) **Return the names of all lakes, rivers and seas.**
- (8) **Return the average length of a river.**
- (9) **Return the name of countries that have more than 10 islands.**
- (10) **Return the length of all rivers in Great Britain.**
- (11) **Return the name of the countries that have the 10 longest total length of rivers.**

**Task 3:** Write SQL expressions that, upon evaluation, returns the data characterised by each of the following English-language specifications. Use duplicate removal where appropriate (e.g., when a duplicate is not required in the intended answer).

- (12) **Return the names of up to 10 countries and the value corresponding to half the country's population.**
- (13) **Return all the information available about cities whose name is Manchester.**
- (14) **Return the name of cities whose name starts with the substring 'Man'.**
- (15) **Return the name of both countries with Buddhist populations and organizations, established after 1st December 1994, that the country is a member of.**
- (16) **Return the name of each country with the number of islands in it.**

For Task 1, you are expected to submit the TRC expressions only. These could just be written in your report (see below). You are not required to run the query against the data as there is no easily available TRC evaluator to use, but if you want to ensure your expression computes the intended result, consider mapping the expression into an iterative computation in your favourite language, as exemplified in the lectures with Python.

For Task 2, you must use the **RA** software<sup>2</sup> to evaluate your expressions and obtain printed-out results for your submission. Use the command line options to read the input from a file and to redirect the output to a file. These files should be part of your submission. Note that **RA** is dependent on **sqlite3** DBMS<sup>3</sup>.

---

<sup>1</sup> <http://www.dbis.informatik.uni-goettingen.de/Mondial/>

<sup>2</sup> <https://users.cs.duke.edu/~junyang/ra/>

<sup>3</sup> <https://www.sqlite.org/>

For Task 3, you must use **sqlite3** to evaluate your expressions and obtain printed-out results for your submission. Use the **sqlite3** commands to read the input from a file and to redirect the output to a file. These files should be part of your submission.

## Marking

- Each of the 16 answers is worth up to 2.5 marks, for a total of up to 40 marks.
- The whole lab is worth up to 40 marks and contributes up to 4 marks to the final mark for the course unit.

## Software/Data

### On Mondial

You can find documentation about Mondial in its website. There are local copies of the ER diagram, the relational schema and a diagram of the referential dependencies in

```
/opt/info/courses/COMP62421/data/Mondial/relational/mondial-ER.pdf
/opt/info/courses/COMP62421/data/Mondial/relational/mondial-RS.pdf
/opt/info/courses/COMP62421/data/Mondial/relational/mondial-abh.pdf
```

### RA

The homepage of RA is:

```
https://users.cs.duke.edu/~junyang/ra/
```

RA uses SQLite as a backend. To use RA you need to copy the jar file

```
/opt/info/courses/COMP62421/software/ra.jar
```

into your working directory, and add a 'properties' file corresponding to the SQLite. For models and examples, see the RA homepage.

You also need the database to be local to you, so that you have write permissions on it. Data is always under:

```
/opt/info/courses/COMP62421/data
```

For this lab work, you will use the Mondial relational data in

```
/opt/info/courses/COMP62421/data/Mondial/relational
```

You should copy the database and the properties files to your working directory

```
/opt/info/courses/COMP62421/data/Mondial/relational/mondial.db
/opt/info/courses/COMP62421/data/Mondial/relational/mondial.properties
```

You can then invoke RA as follows:

```
java -jar ra.jar mondial.properties
```

### sqlite3

The mondial.db file above is used in sqlite3 too. You may benefit from passing the initialization file we prepared (setting parameters for the sqlite3 command-line interface). This is available in

```
/opt/info/courses/COMP62421/data/Mondial/relational/sqliterc
```

If you have copied it into your working directory, you can invoke sqlite3 as follows:

```
sqlite3 -init sqliterc mondial.db
```

For sqlite3, the documentation in the website is very good, but if you prefer learning from books, you have free access (from an UoM IP address) to this one:

The Definitive Guide to SQLite

Grant Allen, Mike Owens

ISBN: 978-1-4302-3225-4

Apress, 2010

<http://link.springer.com/content/pdf/10.1007%2F978-1-4302-3226-1.pdf>