

Part One - Getting the Data Ready

1. Download the following files:

<http://www.epcc.ed.ac.uk/~adam/bbb/file1.txt>

<http://www.epcc.ed.ac.uk/~adam/bbb/file2.txt>

<http://www.epcc.ed.ac.uk/~adam/bbb/file3.txt>

<http://www.epcc.ed.ac.uk/~adam/bbb/file4.txt>

<http://www.epcc.ed.ac.uk/~adam/bbb/capitals.txt>

2. Use `grep` to find out which file has population data in. (Search for the word “population” and remember that you don’t know how it will be capitalised).

3. Rename this file to `population.txt` with a command like

```
mv filen.txt population.txt
```

(you’ll have to replace the *n* with the appropriate number)

4. Use `grep` to find a file about gdp, and rename this `gdp.txt`.

5. Use `grep` to find a file about area, and rename this `area.txt`

6. Use `grep` to find a file about life expectancy, and rename this `life.txt`

The capitals file has some notes in. For the purposes of this exercise, let’s assume that you want to remove these.

7. Type `more capitals.txt` which displays the contents of the file one screen at a time. Press the space bar to advance through the file. You’ll notice that most of the notes have a semicolon “;” in.

You want to check that every occurrence really does consist of a note:

8. Use `grep` to find all of the lines containing a semicolon

9. Use `sed` to remove the semicolon, and everything which follows it on the same line.

10. Use `awk` to remove the first column of the `capitals.txt` file

`mysqlimport` struggles a bit with numerical fields containing “non-numeric” courses, so:

11. Remove the ‘\$’ signs from the GDP file using `sed`.

12. Remove the ‘,’ characters from each of the text files using `sed`.

Part Two - Getting the Data into the Database

0. If you have not already done so, install MySQL. Instructions for doing this are provided in the guide [Installing MySQL for the Practical Exercise](#).

1. Start the MySQL client (as the mysql root user) by typing on to the command line: `mysql -u root`

2. Type `show databases;`

3. Create a new database: `create database bbb;`

4. Set the database that we're going to work with: `use bbb;`

5. Create five tables by entering the following SQL statements into `mysql`¹. Note that SQL reserved words (shown in capitals here as is common practice) are not case sensitive but names of tables and fields are case sensitive:

```
CREATE TABLE population (rank INTEGER, country VARCHAR(50), population INTEGER, date VARCHAR(30));
CREATE TABLE area (rank INTEGER, country VARCHAR(50), area INTEGER);
CREATE TABLE capitals (country VARCHAR(50), capital VARCHAR(50));
CREATE TABLE gdp (rank INTEGER, country VARCHAR(50), gdp FLOAT, date VARCHAR(30));
CREATE TABLE life (rank INTEGER, country VARCHAR(50), life FLOAT, date VARCHAR(30));
```

6. You can check that all of the tables have been created by entering `show tables;`

7. Leave this window open in `mysql` and open a new terminal window

8. Make sure `mysqlimport` is in your path, with a line like `export PATH=$PATH:/usr/local/mysql-5.6.20-osx10.8-x86_64/bin` (you may need to adjust this line to match the directory into which `mysql` was installed)

9. Use the following lines to load your data into the database:

```
mysqlimport -u root bbb population.txt
mysqlimport -u root bbb area.txt
mysqlimport -u root bbb capitals.txt
mysqlimport -u root bbb gdp.txt
mysqlimport -u root bbb life.txt
```

¹ The text-based user interface includes a history mechanism (arrow keys) and tab completion (which will work once table names and field names have been created.)

