Lab exercise 1: Introduction

In order to do the practical parts of this module, you will need access to a MySQL database server and MySQL Workbench. You may use a University system for this, or you may download the software (it is free) and install it on your own computer. If you wish to use your own computer, then notes are provided on the module Moodle site for Windows, OSX and Linux. Part 1 of this exercise assumes that you are using the University system. If you have the software working on your own computer, then you can proceed directly to Part 2.

Useful web resources:

- http://support.cs.nott.ac.uk/help/docs/webpages/
- http://support.cs.nott.ac.uk/help/docs/databases/mysql/using-mysql.html

Part 1 – set up your database account

NOTE: the following instructions will work only from *inside* the University (e.g., via VPN or Windows Virtual Desktop). Direct access to both the MySQL database and MySQL Workbench are blocked by the University firewall but you can access them from outside the University network using the Windows Virtual Desktop (WVD) from home. Instructions for using this are provided at: https://www.nottingham.ac.uk/it-services/connect/working/virtual.aspx

1. Connect to **mersey.cs.nott.ac.uk** (use an SSH client such as PuTTY – which is installed on the lab machines), and log in using your University username and password.
2. From the Unix command line run **setup_mysql**. This will ask you if you want to create a default database for your username (if you hit 'y' a new database with your username will be generated). You can also create a new database with a different name (i.e., select 'c').
3. Choose a name for your database, and make a note of your database username. Your username will be prefixed.
4. Reset the password to something you can remember. NOTE: your database username and password are for this database only – *they have nothing to do with your university username and password*.
5. Run MySQL Workbench and log in by connecting to a new database using the Database menu, specifying the hostname as **mysql.cs.nott.ac.uk**, and the username as provided by the setup_mysql script and the password you set in the previous step.

Part 2 – build your first database

We are going to create and query a database containing some data about 51 countries. You can see the entire dataset in Moodle – the file is linked to **Country data (PDF)**.

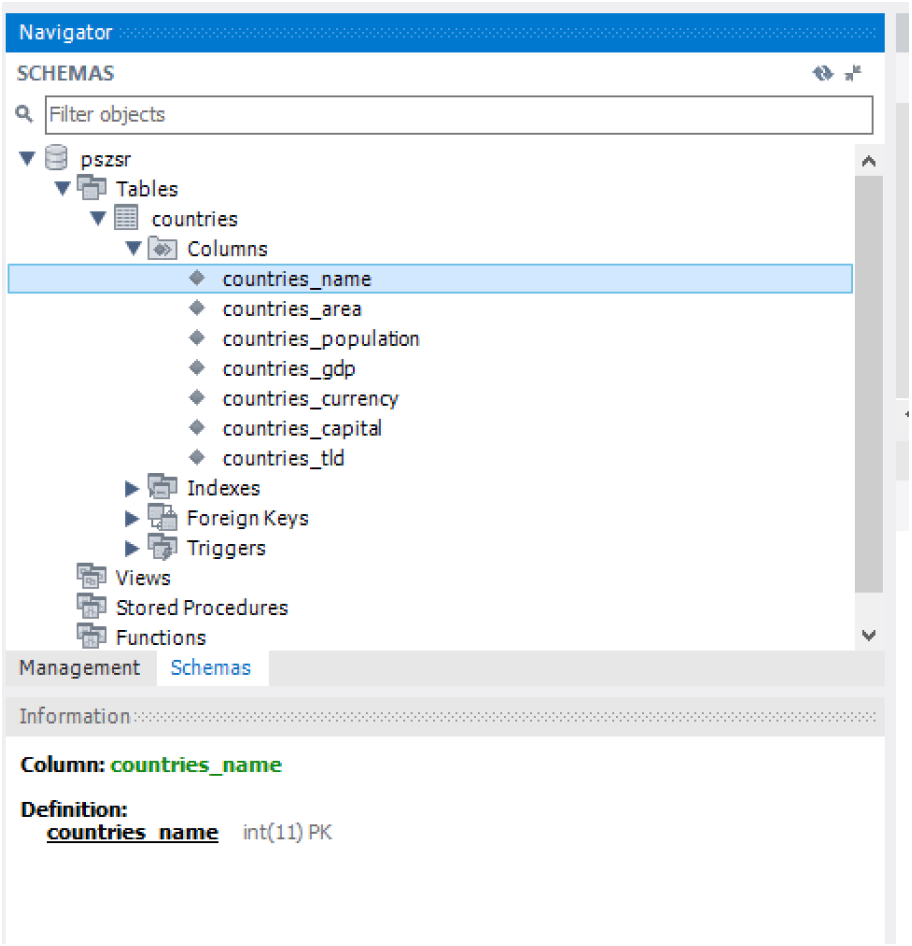
The start of the data is as follows:

Argentina	**2780400**	**43417000**	**5.45866E+11**	**Peso**	**Buenos Aires**	**.ar**
Australia	7692024	24233900	1.20462E+12	Dollar	Canbera	.au
Austria	83879	8725931	3.86428E+11	Euro	Vienna	.at
Barbados	439	277821	4587550000	Dollar	Bridgetown	.bb
Brazil	8515767	206440850	1.79619E+12	Real	Brasilia	.br
Burundi	27834	11178921	3007029030	Franc	Bujumbura	.bi
Cambodia	181035	15458332	20016747750	Riel	Phnom Penh	.kh
Canada	9984670	38286425	1.52976E+12	Dollar	Ottawa	.ca

1. Open MySQL Workbench. If you are using MySQL on mersey then use the database you created with setup_mysql. If you are using it on your own machine then you will probably need to create a database for your work.
2. Create a table called “countries” that has 7 columns.
3. Fill out the form to create the schema of your database. You should create the following fields:

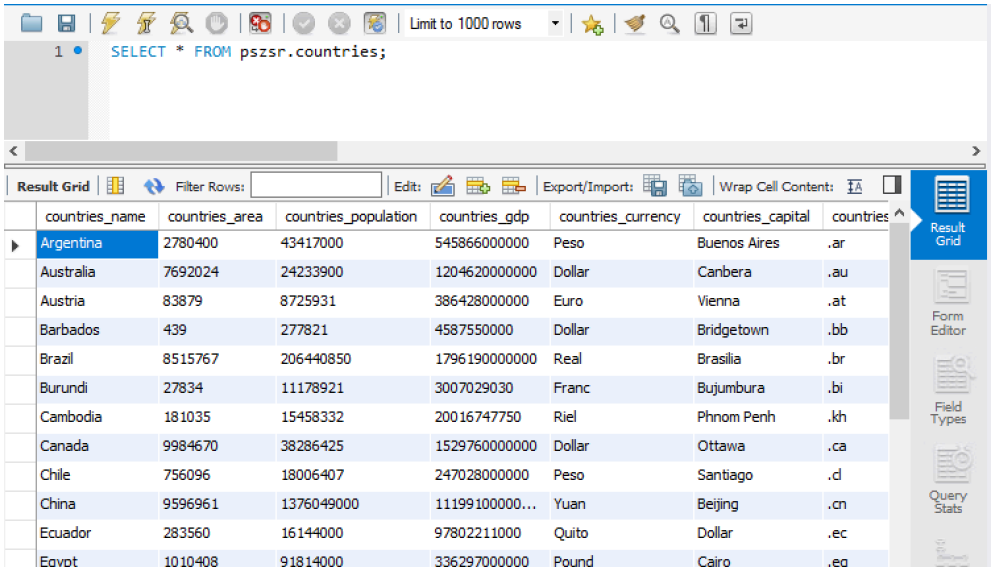
   ```
   countries_name; countries_area;
   countries_population; countries_gdp;
   countries_currency; countries_capital; countries_tld
   ```

   You will need to set the “Datatype” for each field. For the fields that will contain textual data (e.g., countries_name) specify VARCHAR and also specify the maximum length (guess at a value that is larger than anything you are likely to use – we suggest 50 for countries). For the numeric fields (e.g., countries_area) set the type to INT. NOTE GDP tends to be a large number (often in the trillions). Since this might be too big for INT to handle you need to find a data type that can deal with very large integers. Consider using a BIGINT(20) instead.
4. When you have filled out the table (other than Name and Datatype – accept default values for everything) press Apply then Finish. If you select your database in the **Schemas**, you should now see the structure of your table – something like the following:



You are now ready to start entering data.

5. To save you some typing, we have provided you with this data in the CSV interchange file format (you can upload it into MySQL using MySQL Workbench). Download the **Country data (CSV)** file from Moodle and save it on your local machine. Right click the countries table and use the Table Data Import Wizard. Navigate to the data file and select it. Use the existing table you created (i.e., “Use existing table”) option, then check the import settings. If everything is OK, then it should tell you that the input has been successful and that 51 records have been imported.
6. Have a look at the imported data by right clicking the countries table and clicking Select Rows – Limit 1000 for a quick view. It should look something like this:



countries_name	countries_area	countries_population	countries_gdp	countries_currency	countries_capital	countries
Argentina	2780400	43417000	545866000000	Peso	Buenos Aires	.ar
Australia	7692024	24233900	1204620000000	Dollar	Canbera	.au
Austria	83879	8725931	386428000000	Euro	Vienna	.at
Barbados	439	277821	4587550000	Dollar	Bridgetown	.bb
Brazil	8515767	206440850	1796190000000	Real	Brasilia	.br
Burundi	27834	11178921	3007029030	Franc	Bujumbura	.bi
Cambodia	181035	15458332	20016747750	Riel	Phnom Penh	.kh
Canada	9984670	38286425	1529760000000	Dollar	Ottawa	.ca
Chile	756096	18006407	247028000000	Peso	Santiago	.cl
China	9596961	1376049000	11199100000...	Yuan	Beijing	.cn
Ecuador	283560	16144000	97802211000	Quito	Dollar	.ec
Egypt	1010408	91814000	336297000000	Pound	Cairo	.eg

You may also notice that this has generated a simple SQL SELECT query:

```
SELECT * FROM <your username>.countries
```

7. Now check the countries_name record is the primary key. Right click the countries table and select Alter Table. For countries_name, make sure the "PK" checkbox is ticked (NN – NOT NULL – may be checked too). If you change anything, hit Apply.
8. Right click the countries table again and use Select Rows – Limit 1000 to explore your data. Note that you can sort it by any of the fields just by clicking them. You can also edit data values by double clicking on them (make sure you hit Apply, which will then reveal the actual SQL query being used).

Part 3 – query your database

Do the following queries. Firstly, find the population of the Poland. Go to the SQL box and type in the following code and press the Go button:

```
SELECT countries_name, countries_population FROM <your
username>.countries WHERE countries_name = 'Poland';
```

Display the report – it should look something like this:

countries_name	countries_population
Poland	38483957
NULL	NULL

Now try the following exercises:

1. Print the GDP of all countries in the database.
2. Print the GDP of all countries, in descending order (starting with the largest). HINT – you will need to use the "ORDER BY ... DESC" Keyword.
3. Modify this query to print the per-capita GDP of all countries in the database, in descending order (starting with the wealthiest). HINT – you will need to do

an arithmetic calculation in your query. NOTE – this should show Luxembourg at the top of the list, and Burundi at the bottom..

4. Modify this query to display 2 places of decimals. HINT – you will need to use the ROUND function in your query.
5. Modify this last query so that it ignores very small countries with a land mass of less than 20,000. HINT – use the WHERE keyword. NOTE – this should now show the list without Luxembourg, Singapore and Qatar, with UAE at the top.
6. Write a query that shows the names of countries that have a land mass in between 100,000 km2 and 200,000 km2. HINT – use the BETWEEN ... AND keyword. NOTE – this should now show only Cambodia, Greece, Malawi and South Korea.
7. Write a query that shows the names of countries that start with the letter ‘M’. HINT – use the LIKE keyword with the % wildcard.
8. Write a query that lists the names of countries that end with ‘...land’.
9. Write a query that shows the countries, where the name of the country is the same as the name of the capital.
10. Find the total population of all of the countries in the database. HINT – use the SUM function).
11. Find the number of countries that have a total land mass of at least 500,000 HINT – use the COUNT function.

And lastly, if you have done all of the above, here is a much more tricky one:

12. Show the countries which have a per capita GDP more than that of Poland, in descending order of wealth. HINT – you will need to use two SELECT statements, feeding the output of one into the other. This is called nesting.