Lab 1 – DD1334 – Getting Access to the Database

In this lab you will install and familiarize yourself with SQLite – a very simple to use and install SQL database management software that we will utilize in this course.

You will interact with the database via the command line by executing the command line program **sqlite3** after installing it on your own computer or alternatively using a remote KTH server called student-shell.sys.kth.se.

Installation of SQLite:

Option 1: Install on your own computer:

- If you use a Mac, sqlite3 should be pre-installed and you can most likely skip this step.

  Otherwise, a detailed instruction about how to install sqlite3 can be found for example here: https://www.servermania.com/kb/articles/install-sqlite/

- To test that it works, open your computer's terminal/command line and execute the program "sqlite3" as described in the linked article above. Type **.quit** followed by enter to exit sqlite3.

Option 2: Use the pre-installed sqlite3 via the student-shell.sys.kth.se server instead:

- For this you need to first get access to the kth unix server **student-shell.sys.kth.se**. You can do this using an SSH shell. MacOS and Linux/Unix typically have an SSH command line client pre-installed. Otherwise search online for "Install OpenSSH" to install the SSH software to connect to the remote machine. You can also use the computer lab computers if you don't have a laptop/PC available.

  Help with KTH Unix, logins, etc can also be found here: https://intra.kth.se/en/it/arbeta-pa-distans/unix

- Open a termial window and type **ssh yourusername@student-shell.sys.kth.se** where yourusername is your CSC username (typically your kth email account has the form yourusername@kth.se and the password is the same as your KTH password. You are now logged in remotely on student-shell.sys.kth.se, a linux server. Type "sqlite3" and you will be again inside the SQLite interactive shell. Type **.quit** followed by enter to exit sqlite3.

Downloading our Sample Database for this lab:

- **If you use sqlite3 on your own PC:**
  Create a new folder "lab1" on your computer. Download the file "mondial.db" by clicking on download at this URL and saving the resulting mondial.db file in your "lab1" folder: https://github.com/seralf/mondial_sqlite/blob/master/db/mondial.db

  If you are using sqlite3 on student-shell.sys.kth.se instead of your own PC:

  open a terminal from the computer labs or your own PC, type
  "ssh yourusername@student-shell.sys.kth.se" then type
  "mkdir ./lab1" to create your directory followed by
  "cd ./lab1" to change directory into "lab1" and then execute the command:
  "wget https://github.com/seralf/mondial_sqlite/blob/master/db/mondial.db?raw=true -O mondial.db"

  to download the mondial.db file on the kth server.

Getting familiar with the sqlite3 command line:

- Open a terminal and change your working directory to the “lab1” folder (you can use use “cd ./lab1” in unix)
- Type “sqlite3 mondial.db” and press enter. You now have loaded the database and are inside the sqlite interactive shell where you can execute SQL commands one by one.
- The following commands are worth trying to get started:
    - .help: lists basic sqlite specific commands (commands starting with a dot are SQLite specific commands and not standard SQL)
    - .tables: shows all available database tables/relations
    - arrows up/down: go through previously executed commands
    - select * from cities limit 5;
      will display 5 entries from the cities relation in mondial.db – note the “;” at the end. If you don’t type “;” and press enter you will be creating a new line in the current command without executing it until you end with “;” followed by enter.
    - “.mode column” changes result display of queries to pretty column formatting. If this was not the case before, try executing “select * from cities limit 5;” again – it should now be much more readable.
    - “.header on” displays the column names when displaying search results
    - You can also create a config file with you display settings if you don’t want to re-execute .mode column etc every time you restart sqlite3. For this, you create a text file with the name “.sqliterc” in your home folder and the file should contain the commands you want executed at startup, for example two lines:

      ```
      .header on
      .mode column
      ```

Lab 1 – How to submit solutions & get help

Please check the times for our lab sessions on Canvas. You will need to pass the lab by the date stated on our overview page to get an “A” and the grade decays thereafter with delays to submission.

Together with your lab partner, you can then book your short lab session help or examination appointment time slot with one of the tutors on canvas. You can use a booked lab slot either to just ask questions about labs you are currently in the process of working on, or to present your lab solution. Since these time slots for you, your lab partner and with the tutors are only a few minutes in length, **you need to be well prepared**. Also, do not book multiple individual time slots in the same lab session.

For presenting your solution, you in particular need to

- make sure you have **reserved a lab session time-slot** with one of the available tutors for you and your lab partner on canvas. Instructions: https://community.canvaslms.com/t5/Student-Guide/How-do-I-sign-up-for-a-Scheduler-appointment-in-the-Calendar/ta-p/536 Please make sure you only book one time slot with your lab partner for that – you do not need two individual time slots.
- upload your solution files to canvas **at the latest the day before the lab session** during which you want to present
- **Log into our lab session zoom video call** (see link on canvas) a few minutes before your individual lab session time slot. You will be put into the “waiting room” and the tutor will move you into an individual call with your lab partner at your assigned time slot. If you miss your timeslot, you may have to wait until the next lab session if no other time slots are free, so please be on time.
- **have all material of your solution ready to present via screen sharing** and be ready to run your code/solution live.
- **be ready to answer any additional questions** the tutor may have about your solution

Lab 1 – DD1334 – Questions

Note: You can complete the lab alone or with one other student from the course. To pass the lab, you need to present your solutions in one of the lab sessions with your lab partner. Please note the deadlines and grading scheme depending on the timeliness and correctness of your submission described on the course Canvas page.

Important: Besides getting the results indicated below you and your lab partner may be asked by the TAs to explain what is happening and also solve some simple similar queries on the spot to show that you have understood the material!

Goals: This lab focusses on getting you familiar with real world SQL. This lab just focusses on setting things up and some basics. In Lab 2, we will expand on more complex queries and data science questions using the same database. You should expect Lab 2 to take more time to complete than Lab 1.

Before you start, please read Chapter 6.1 “Simple Queries in SQL”, Chapter 6.2 “Queries Involving More Than One Relation”.

Imagined Scenario: Welcome, this is your first day as junior data analyst at the UN. You have been given access to the mondial database of country facts. Your team is preparing a slide deck for an upcoming presentation and needs urgent answers to the following questions. Your boss didn’t specify what structure the database has but just said “here are the login details for our database”.

First, you need to get an overview of what’s going on. Use the commands “.tables” to list the available relations and “.schema somerelationname” to look at the schema for a table as an SQL output. Alternatively you can use the command “pragma table_info(“somerelationname”);” to present the result in a more visual manner. Don’t forget that the special sqlite3 commands that start with “.” SQL commands are terminated with a “;”, otherwise you will move into the next line where you can continue entering the command (useful for multiline complex SQL queries). If you press enter before the “;”, just enter “;” in the new line followed by the enter key to terminate the current command.
Here is the list of relations that you will see, but for this lab you will work only with **city** and **country** though:

```
sqlite> .tables
Airport            EthnicGroup        Provinceothername  geo_Lake
City               Island             Provpops           geo_Mountain
Citylocalname      Lake               Religion           geo_River
Cityothername      LakeOnIsland       River              geo_Sea
Citypops           Language           RiverOnIsland      geo_Source
Continent          Mountain           RiverThrough       isMember
Country            MountainOnIsland   Sea                islandIn
Countrylocalname   Organization       borders            located
Countryothername   Politics           encompasses        locatedOn
Countrypops        Population         geo_Desert         mergesWith
Desert             Province           geo_Estuary
Economy            Provincelocalname  geo_Island
```

You can use the command **SELECT * from relationname LIMIT 5;** to see 5 tuples in the relation to get a sense of what is stored. The way in which different parts of the database relate to each other are mostly self explanatory, but you can go to https://www.dbis.informatik.uni-goettingen.de/Mondial/ to read more about the schema. Also, if you have not created a “.sqliterc” startup file don’t forget that executing “.mode column” will enable pretty printing of results. Also it is a good idea to execute “.header on” for ease of interpreting the results.

Questions:
Create SQL Queries answering each of the following. We will actually only need the relations: **city, country** for now. The answers that you should get are also indicated.

1) Write any query that allows you to quickly determine the countries with a population of less than 1000?

   Answer should be: Holy See, Cocos Islands, Pitcairn

2) Now be more precise and write an SQL query that returns a relation that only has attributes “CountryName”, “Area” and “Population” to answer the previous question. Note that you will need to rename the attribute “Name” in the table to “CountryName” here.

Answer should be (note that the output name for Cocos islands is truncated due to default display width only you can set “.width 20” to have 20 character default width for the first output column to see the full name):

```
CountryName  Area        Population
-----------  ----------  ----------
Holy See     0.44        842
Cocos Islan  14          596
Pitcairn     47          56
```

3) List the attributes “Name, Population, Area” for all the countries that have a population of between 1000 and 2000 and whose area is at least 1.0 (km^2)

Answer should be:

```
Name        Population  Area
----------  ----------  ----------
Svalbard    1872        62049
Niue        1611        260
Tokelau     1383        12
```

4) What is the country code for Norway in the relation ‘country’? Write an SQL Query.
Answer should be: ‘N’

5) What are the cities in Sweden with population larger than half a million? Write an SQL query.
Answer should be: Göteborg, Stockholm

6) Produce a relation containing only city names, populations and elevations for those cities lying below 0 metres in height above sea level.

Answer should be:

```
Name        Population  Elevation
----------  ----------  ----------
Astrakhan   527345      -28
Lelystad    76252       -3
Almere      196244      -3
Babol       250217      -2
Baku        2150800     -28
Atyrau      196494      -20
Aktau       181526      -8
New Orlean  343829      -2
David       144858      -6
Georgetown  118363      -2
```

7) What is the sum total, average, minimum and maximum population for cities below 0 metres? Write a single query for this with output column names “total, average, minimum, maximum”

Answer should be:

```
total       average     minimum     maximum
----------  ----------  ----------  ----------
4185928     418592.8    76252       2150800
```

8) Find all city names whose name starts with “Los” or ends with “holm” but that do not end with “is”. First write a query for this that combines two queries using the ‘except’ keyword. Then write another query with the same result that does not use ‘except’ but instead a single more complex boolean expression following the ‘where’ keyword. Do brackets matter here?

Answer should be:

```
Name
-----------
Los Angeles
Los Teques
Stockholm
```

9) Return a list of country names and populations of the 5 largest countries by population (hint: use LIMIT).

Answer should be:

```
Name        Population
----------  ----------
China       1360720000
India       1210854977
United Sta  318857056
Indonesia   252124458
Pakistan    207776954
```

10) Find the lowest altitude 5 cities in increasing altitute and display their names and city elevations as well as country names (in long form, not just the country code). Do not display results for cities that do not have a valid numerical elevation.

Answer should be:

```
Name        Name        Elevation
----------  ----------  ----------
Russia      Astrakhan   -28
Azerbaijan  Baku        -28
Kazakhstan  Atyrau      -20
Kazakhstan  Aktau       -8
Panama      David       -6
```

11) Produce a relation with a single attribute ‘name’ and which contains the names of either countries starting with ‘Y’ or cities ending with ‘x’.

Answer should be:

```
Name
----------
Bordeaux
Halifax
Jizzax
Phoenix
Sfax
Yemen
```