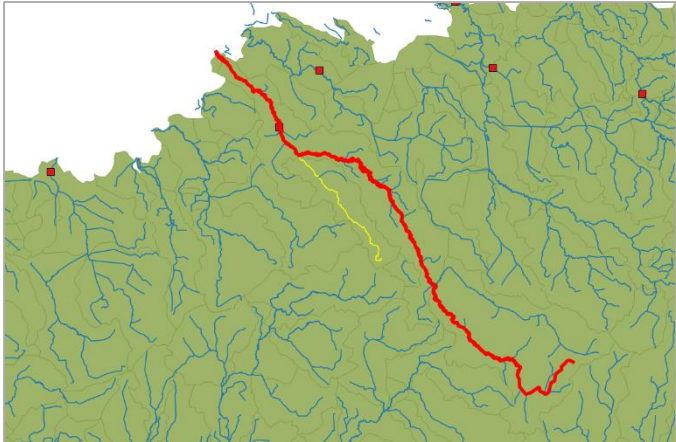


An Introduction to Databases – practice task

The objective of the task is to familiarize yourself with the basics of databases and how to work with them using C.

```
Host: ewis.pld.ttu.ee
Database: riverdata
Port: 5432
Username: student
Password: iag0582
```

river	station
id_river	id_station
id_outlet	id_river
id_mainriver	name
id_watershed	alias
id_subwatershed	dist_from_outlet_km
subwatershed_name	basin_km2
name	l_est_lon_x
type	l_est_lat_y
outlet_phase	dist_from_outlet_m
type	comment
outlet_phase	
dist_from_outlet_km	
length_km	
lon1	
lat1	
lon2	
lat2	



You will be using two tables in the database: a) **river**, which contains all the Estonian rivers and b) **station** which contains measurement stations located by the rivers. The attributes of interest in the first table are **length_km** specifying the length of the river in kilometres and the geographical coordinates of the starting and ending point of the river (**lon1, lat1** – beginning; **lon2, lat2** – end). Each station is related with a river that it is located by using the key **id_river**.

Tasks

1. Write a C program that would connect to the database given above, queries 5 longest rivers and outputs it on the terminal screen.
2. Query all rivers that are longer than 70 km and which are considered 'main rivers' or in other words, rivers that do not flow into another river. Use *id_mainriver* to identify whether the river is a 'main river'.
3. Find all the stations that are located on the rivers found in the last task. (using JOIN)
4. Query the start and end points of the rivers from the database and find the distance between them.

To find the distance between two points you can use the following formula:

$$d = R * \text{acos}(\sin(\text{lat1}) * \sin(\text{lat2}) + \cos(\text{lat1}) * \cos(\text{lat2}) * \cos(\text{lon1} - \text{lon2}));$$

where R is the radius of the earth. R=6371 in km. Also keep in mind that the coordinates are in degrees, but trigonometric functions take inputs in radians.

Helping materials

Postgre documentation - <http://www.postgresql.org/docs/9.1/interactive/index.html>

Using the Postgre libpq C library - <http://www.postgresql.org/docs/9.1/interactive/libpq.html>

- It is recommended to use class computers as the libraries are already pre-installed.
- To add Postgre libpq library
`#include <pgsql/libpq-fe.h>`

When compiling the program you need to add an extra parameter `-lpq` to link the library. For example:
gcc main.c -lpq

- For doing the calculations in the last task you will probably also need math library `math.h`:
`#include <math.h>`

While compiling you will need the parameter `-lm` to link the math library. For example:
gcc main.c -lpq -lm

- Main functionality of libpq

```
//Creating a connection
PGconn *conn;
conn = PQconnectdb("dbname=<database name> host=<hostname> user=<username>
password=<password>");

//Verify the connection
if (PQstatus(conn) == CONNECTION_BAD) {
    //If connection was unsuccessful
    /*Error handling*/
}

//Executing a query. For example let us query 10 first rows of the table specified.
PGresult *res;
res = PQexec(conn, "SELECT * FROM <name_of_the_table> LIMIT 10;");

//Verify that result was received
if (PQresultStatus(res) != PGRES_TUPLES_OK) {
    //No rows were returned
    /*Error handling*/
}

//Finding the number of returned rows
int row_count = PQntuples(res);

//Querying the value from the specified row and column
char *val;
val = PQgetvalue(res, 0, 1); // will return the value from row 0 and column 1

//free the PGresult handle
PQclear(res);

//Disconnect and free the Pgconn handle
PQfinish(conn);
```

Example code - <http://ati.ttu.ee/~hkinks/iag0582/praktikum11/psql.c>

compilation: `gcc psql.c -lpq`

Advanced tasks

1. Include **measurement** table, that contains *value* field with measurement data about the amount of water flowing through the river in a second (mass flow rate of river in units m^3/s). Find 5 rivers that have the highest average mass flow rate.
2. Find and output the yearly averages of the 5 longest rivers during the timeperiod 1990-2000 (in total 50 rows).
3. Are the volumetric flow rates correlated for different rivers? Based on the query made in the last point find the correlation coefficient between the yearly measurement averages of *Pärnu* and *Võhandu* river using the CORR(X, Y) function.
4. Using the fields *id_river*, *id_outlet* and *id_parent* output rivers in a hierarchical manner. Limit the output by asking the “main stem”. For example:

```

Choose river: Pärnu
-Pärnu
  +Navesti
  -Sauga
    +Hirve
    -Are
      -Kõrisoo
      +Saki
        -Tõntsu
        -Marjassaa
          +Elbu
          ...
  
```

