

Sissejuhatus andmebaasidesse – tunniülesanne

Tunnitöö eesmärk on tutvuda

The objective of this task is to get some basic knowledge about databases and how to work with them using C. A PostgreSQL database has been set up for this purpose:

```
Host: ewis.pld.ttu.ee
Andmebaas: ablid
Port: 5432
Kasutaja: student
Parool: iag0582
```

The database consists of three tables, however in this lesson we are going to use only the table named “**joe_andmed**”. The table contains Estonian rivers, including their length (**pikkus_km**) and source (**lon1**, **lat1**) and end (**lon2**, **lat2**) point geographical coordinates.

joe_andmed		
id_jogi	int4[10]	
id_suubla	int4[10]	
id_peajogi	int4[10]	
id_vesikond	int4[10]	
id_avk	varchar[6]	
avk_nimi	varchar[13]	
joenimi	varchar[15]	
tyyp	int4[10]	
suubjark	int4[10]	
suudmest_km	numeric[4,1]	
pikkus_km	numeric[4,1]	
lon1	numeric[8,6]	
lat1	numeric[8,6]	
lon2	numeric[8,6]	
lat2	numeric[8,6]	
< 0	1,755 rows	0 >

Tasks

1. Write a database interface in C, which connects to the aforementioned PostgreSQL database and outputs 5 longest rivers of Estonia.
2. Calculate the distance of source and end point for each river.

For finding the distance, use the following formula:

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

Helping materials

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

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

- It is recommended to use ICT-502 computers as the PostgreSQL libraries are already there and set up for use.
- To include the libpq C libraries, your program must have **#include <pgsql/libpq-fe.h>**

Also, when compiling, the libraries have to be linked with a parameter *-lpq*. For example:
gcc main.c -lpq

- To include math libraries:
#include <math.h>

And compile with parameter *-lm*. I.e:
gcc main.c -lm

- Main functions of libpq

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

//Check connection status
if (PQstatus(conn) == CONNECTION_BAD) {
    //Connection failed
    /*Error handling goes here*/
}

//executing a query, for example select 10 first records from <table>
PGresult *res;
res = PQexec(conn, "SELECT * FROM <table> LIMIT 10;");

//Check query result status
if (PQresultStatus(res) != PGRES_TUPLES_OK) {
    //No results returned
    /*Error handling goes here*/
}

//return results count
int row_count = PQntuples(res);

//get value from specified row and column
char *val;
val = PQgetvalue(res, 0, 1); //returns value from row 0 and column 1

//Free PGresult handle
PQclear(res);

//Close the connection and free the memory used by PGconn handler
PQfinish(conn);
```

- Sample code - <http://ati.ttu.ee/~hkinks/iag0582/psql.c>
Compiling: `gcc psql.c -lpq`