# Assignment #5: Database Programming

# ECE 650 – Spring 2016

**Due by 11:59pm eastern time on Wednesday, April 20**

## General Instructions

1. You will work individually on this assignment.
2. The code for this assignment should be developed and tested using a Linux Virtual machine that you may create at the following location:
   https://vm-manage.oit.duke.edu/vm_manage
   You must select following image: vcl-ubuntu14-generic
   Other environments, unfortunately due to complexity, will not be able to be supported.

3. You must follow this assignment spec carefully, and turn in everything that is asked (and in the proper formats, as described). Due to the large class size, this is required to make grading more efficient. Particularly for this assignment, much of the testing will be automated. If you do not follow exact instructions for your submission materials (file names, program output, etc.) points will be deducted.

## Overview

In this assignment, you will build a MySQL database. The relation schemas (tables) of the database will be provided, along with the database values. You will implement:

1. A C program to read text files containing the entries (rows) for each table, and build the database by creating each table and adding entries.
2. A C library (set of functions) which will provide an interface for a program to interact with the database (e.g. add rows to the tables and query for certain information).

The database relates to ACC basketball teams, and will allow queries to discover information about things like player statistics, team attributes, etc. The database will have 4 tables, specified as follows (an underlined attribute indicates the primary key for the table):

PLAYER (PLAYER_ID, TEAM_ID, UNIFORM_NUM, FIRST_NAME, LAST_NAME, MPG, PPG, RPG, APG, SPG, BPG)
TEAM (TEAM_ID, NAME, STATE_ID, COLOR_ID, WINS, LOSSES)
STATE (STATE_ID, NAME)
COLOR (COLOR_ID, NAME)

*Note the following abbreviations: MPG = minutes per game, PPG = points per game, RPG = rebounds per game, APG = assists per game, SPG = steals per game, BGP = blocks per game

## Tips on Working with the Virtual Machine

When you create your virtual machine and log-in for the first time, you will notice there may be few programs installed (e.g. no gcc, emacs, vim, etc.). You can download your choice of software easily using the command: sudo apt-get install <package name>. For example:

```
sudo apt-get install gcc
sudo apt-get install emacs
```

In order to create a MySQL database and interact with the database through a C API, you will need to install the following two packages:

```
sudo apt-get install mysql-server
sudo apt-get install libmysqlclient-dev
```

## What You Will Implement

Several skeleton files, and database table information files are provided to get you started. The following describes each file:

- **Database source text files: player.txt, team.txt, state.txt, color.txt**. These files contain table-like information for each entry and each attribute that should be inserted into the respective database table.
- **main.c**: The main function. Here you should implement code which will setup the database on each execution of the program. Specifically, it should drop (if needed) and

add each table to the database (named ACC_BBALL), and then read information from the source text files and add rows to each table as appropriate.

- **query_funcs.h and query_funcs.c**: Here you will implement functions to interact with the database (add new entries and print the results of 5 different queries specified below). You may add new functions to these files if desired (you don't have to), but you should not change the definitions of the existing functions.
- **exerciser.h and exerciser.c**: Here you can add code in the exercise() function to test your query functions. The exercise() function is called from main() after the database is initialized.
- **Makefile**: Will compile all source files into an executable program named "test"

Your task is to do the following:

- Create a MySQL database named ACC_BBALL
- Create a user for the ACC_BBALL database named "myuser" with password "passw0rd"
  - mysql> CREATE USER myuser@localhost IDENTIFIED BY 'passw0rd';
  - mysql> GRANT ALL ON ACC_BBALL.* to myuser@localhost;
- Implement support in main.c to initialize to connect (as user 'myuser) to the ACC_BBALL database and initialize its tables and data. Of course, usually a database will live persistently, but for the purposes of evaluating the submissions, the program should re-create the database every time.
- Implement the following query functions:
  - query1(): show all atributes of each player with average statistics that fall between the min and max for each enabled statistic
  - query2(): show the name of each team with the indicated uniform color
  - query3(): show the first and last name of each player that plays for the indicated team, ordered from highest to lowest ppg (points per game)
  - query4(): show first name, last name, and jersey number of each player that plays in the indicated state and wears the indicated uniform color
  - query5() show first name and last name of each player, and team name and number of wins for each team that has won more than the indicated number of games
  - **Important Note:** Each query function should print its output. The format of this output should be as follows. The first row of output should contain each field name (separated by a single space character (see example.3.c). The next rows of output should contain the values returned from the query, each also separated by a single space. The field name and each row of output should appear one line after the next.
- You may test your database and query functions by adding calls to the query functions inside the exerciser() function. For the purposes of grading assignments, we will replace the exerciser.c file with a new one that will test a variety of query calls.

## Detailed Submission Instructions

Your submission will include the following files:

1. **Source Code Files:** main.c, query_funcs.h, query_funcs.c, exerciser.h, exerciser.c
2. **Makefile – Even if you have not made changes to the one provided**
3. **Database source text files**: player.txt, team.txt, state.txt, color.txt

You will submit a single zip file named “hw5.zip” to your sakai dropbox location, e.g.:

```
zip hw5.zip Makefile *.h *.c *.txt
```