# Project 7: Client-Server Databases

**Objectives**
Understand how a common high-level language database API works.
Using a language of your choice, learn how to write a simple program that acts as a simple MySQL client that connects to a MySQL server.

**Overview**
Using the language of your choice, and a complementary MySQL library, create a simple program that successfully connects to a MySQL server hosted on AWS. The program should merely:

- authenticate with the server
- retrieve data via a simple SELECT
- display the data retrieved
- close the connection and quit

Here is a simple example interaction:

```
Snacks database connectivity test.
Connecting... connected!

foo
bar
baz
booyah

Test complete, exiting.
```

The program may then terminate.

**Instructions**
First, you must create yourself an account within the database server, and ensure that your user account has appropriate permissions on the `snacks` database hosted by the server. To do this, you need to connect to the server by running the `mysql` client on the server.

But in order to do *that*, you'll need, at minimum, the `mysql` client. First, recognize that you can download and install MySQL Server "Community Edition" at no charge. But if you do not want to install all of MySQL, you can just install the client.

While I encourage you to use the command-line client, you may also try a GUI tool of your choice, such as Navicat or SQLYog.

**Windows:** Do not use the MySQL installer unless you intend to install MySQL server. Download and extract the zip (do not use the installer!) and locate the `mysql.exe` program in the `bin` folder. Place it somewhere sane (eg C:\Program Files\mysql) and be sure to add its location to your PATH.

**Linux:** Note that the CSCI403 VM and on-campus Linux machines (eg, alamode) have the MySQL client & server already installed. Use a package manager to install `mysql-client` (or whatever your fancy package manager calls the mysql client package). Or download and install from source.

**OSX:** Try `brew` or download and install using the dmg archive.

Next, you should be able to connect to the remote MySQL instance:

```
mysql -h csci403.c99q7trvwetr.us-west-2.rds.amazonaws.com -P3306 -u ybakos -p
```

And use the password provided in class.

**Tip:** As soon as you face an issue, be sure to raise the problem on Piazza. Include the exact command you are executing and the exact output you see.

You should now be logged in as a "root" user on the MySQL server. As such, you can create users and GRANT/REVOKE permissions. Create a user for yourself that has appropriate read/write access to the database called `snacks`. [handy url]

```
CREATE USER 'jsmith' IDENTIFIED BY 'goodpwd';

GRANT ALL ON snacks.* TO 'jsmith';

FLUSH PRIVILEGES;
```

You have just created a new user who can connect to the MySQL server from anywhere, and have granted this user ALL permissions on the `snacks` database.

Try this:

```
USE mysql
SELECT user, host FROM user;
SHOW GRANTS FOR jsmith;
```

Now exit the MySQL session and create a new session using your own user account:

```
mysql -h csci403.c99q7trvwetr.us-west-2.rds.amazonaws.com -P3306 -u jsmith -p
```

And enter the password for your account. You should be able to `use snacks` and `show tables` in the `snacks` database. Do not proceed until you can do this.

Once you have accomplished creating your own account, do not log in as the root user -- use your own account credentials from here on out.

**Create a Simple Program**
Every popular language has a MySQL library available, which allows you to create database-backed applications on a variety of platforms.

Using the language of your choice, now create a simple program that connects to our remote MySQL instance using your credentials, specifies the database to use (`snacks`), and retrieves the results of:

```
SELECT name FROM project_test;
```

See the *Overview* at the start of this document for some sample output.

You will need to explore or discuss recommended database drivers for your language, and explore how to use that library's API to communicate with the MySQL server. One thing is certain: the concepts of configuration, connection, querying and closing should all be familiar.

**Presentation**
Good writing is important, **especially when it comes to code**. *Writing with poor readability, lack of appropriate whitespace, misspellings, etc will be handed back for re-writing (no grade penalty).*

Submit a zip of your code via blackboard. I would prefer a zip containing only *one source file* rather than a bunch of project assets and the like. Keep the code in this assignment simple (think "script" and not "full-blown OO application stack").

**Grading Criteria (90 points)**
Successfully create database user for yourself (30 points).
Complete implementation of a simple program that communicates with the remote MySQL server (60 total points).
- writing quality (30)
- implementation (30)

*Please submit your work via Blackboard by 7AM on the due date.*