# COMP 1536 – Lab 8

**Due:** Two weeks from when it’s assigned, at the beginning of class

**This assignment is to be done individually only.**

# Part A

Create a simple database and populating it with some sample records. To be demonstrated in class.

We will be using MYSQL and familiarizing ourselves with the terminal (i.e., console). The terminal is a command interface where you type commands into it. It will be quicker to use this rather than the web interface in MYSQL.

### OSX

In Spotlight (the search field on the OSX menu bar, top right), type: terminal

You will get a command window where you can type commands.

### Windows 10

Go to search bar, type CMD or PowerShell. You will get a command window where you can type commands.

## Step 1

With MYSQL already installed and running, you’ll want to add the ‘mysql’ command to the terminal.

### OSX

In OSX, go to Spotlight and type ‘terminal’. This will bring up a terminal windows (AKA, command or console window). Having a terminal allows you to type commands. We first need to ensure that the terminal windows that we are using knows about the MySQL program so that when we type ‘mysql’ in the terminal, it knows to go into MySQL.

Do this by typing the following in the terminal:

PATH=$PATH:/Applications/MYSQL/Library/bin

export PATH

Now you can – at any time – get into MySQL via the terminal.

### Windows 10

Follow this tutorial:

<https://www.ghacks.net/2015/09/22/microsoft-improves-environment-variables-editor-in-latest-windows-10-build/>

To set your environment variables.

Now type:

mysql –u root -p

And press enter. Then type ‘root’ for the password. This will get you into MySQL. The default install for MySQL in MYSQL uses root as both the user name as well as the password. Not terribly secure but since it’s just for testing on our computer it should be alright.

## Step 2

Create a database by typing the create database command in MySQL and then use that database. In this example, we’ll call it ‘test’:

create database test;  
use test;

**Note**: most installs of WAMP/MYSQL/LAMP/XAMP may already have a database called ‘test’ we can use the test database or create a new database. You can always type

## Step 3

You can type the commands to create a database but then you’d have to retype the commands every time. Or, you can create a text file and upload the text file all at once. This saves you having to retype over and over again.

Create a database using the commands shown in lecture and put the commands into a text file (Note: use a text editor – which means that the program must save as text, not binary). Create the following:

* Two tables:
  + Where the first table is called customer and has a(n):
    - Primary key
    - First name
    - Last name
    - Email address
  + Second table is called address and has a(n):
    - Primary key
    - Postal code
    - Street name
    - Number
    - Province

Also create dummy data for at least 10 records in each table

Using DDL statements (as shown in the lecture) type your commands in a text file. When you are finished, name your text file ddl.txt, then type:

mysql –h localhost –u user -p test < ddl.txt

This assumes the DB is called test and a text file called ddl.txt.

**Note:** You’ll need to ensure that the terminal path is in the same directory as the text file you save in (e.g., Downloads).

You can check your progress using the commands shown in lecture to display the database and its tables.

## Step 4

Create another text file and call it data.txt. This file will contain your insert and update statements. Once you’ve populated several records (e.g., five), save the text file and type the following at the terminal:

mysql –h localhost –u root -p test < data.txt

If everything goes well, you will be able to see the progress of your database and its tables by performing a select statement and using the asterisk as shown in lecture.

Type quit within the terminal.

## Step 5

It’s nice to be able upload data and DDL statements into the database, but what about extracting the data and DDL statements for backup purposes? There is a command to do this: mysqldump

In order to save the DDL statements, type the following command:

mysqldump –-no-create-info test > snapshot-data.txt

In order to save the data from all tables in your database, type the following command:

mysqldump -–no-data test > snapshot-ddl.txt

This allows you to quickly upload and download all data and DDL for your database.

Take a look inside of these files that you just created.

Note: you’ll see a slightly different form of both your DDL statements as well as your data. That is okay; it is just MySQL giving you specific and extra data based on its database syntax and configuration.

# Part B

For this exercise you will be creating a database in MySQL and accessing the tables from it by looping through it via JavaScript in Node and displaying it within a web page by making an AJAX call (just like the previous lab). Taking the data that you created with part A.

Your job is to write fill in the code in both the JavaScript on the client side as well as JavaScript in Node and have it perform the required update within the page using AJAX with JQuery. So, your task list is:

1. Create the try/catch block of code to call up the database and access both tables from the database and place them both into the divs listed below
2. Fill in the code in JavaScript in Node that both returns the data from the tables as HTML as well as JSON
3. Accept the data as JSON and turn that data into a table and put that table into the element whose id is ‘jsonTable’
4. Accept the data as HTML and turn that data into a table and put that table into the element whose id is ‘htmlTable’
5. Both tables need to be styled with a border, background color, and every second row a different color. Also ensure that the header at the top is styled with bold text and a different color than the rows of data in the rest of the table

**Note: you must use loops for both your JavaScript as well as your JavaScript in Node – do not hard code array indices into your code; you will lose marks for doing so.**