Table of Contents

1. Installation Ideas

For this class you should be working with a local installation of MySQL I am assuming a default installation of MySQL 5.5 or higher with InnoDB tables and strict mode. I assume that you are not trying to do your assignments and experiments on a production server; the things that I ask you to do in this class should not cause problems but students do occasionally make mistakes that could cause problems and you do not want to do that on your job.

For this class you need to have your own local installation of MySQL. I cannot provide individual installation help since people have different systems. If you have not yet installed the MySQL software, you can go to the MySQL page and get the software. You probably want to install the most recent Generally Available (GA) Community Server version (5.6) . If you have version 5.5, that is OK for the class. Start at

```
   http://dev.mysql.com/downloads/mysql/
```

You can get some installation help from the MySQL documentation (chapter 2 Installing … MySQL). You might also find web pages that show you screen shots of an installation for your OS.

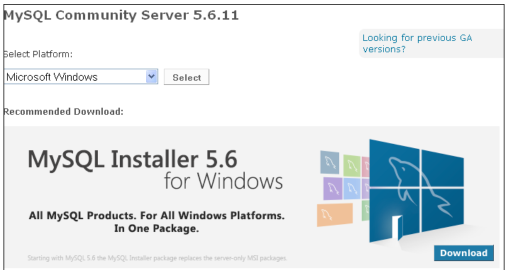
You should always set a password for the root account. And you need to write down that password someplace secure.

There is info in the book and notes about setting up other user accounts. It helps to know where on your system the software is installed and where your data files are stored- but you can generally accept the default locations.

Macs: This is a link provided by the Mac people in a previous semester- that other people thought was well written.

http://www.macminivault.com/mysql-mountain-lion/

Windows : You can generally use the installer program that Mysql provides. If you have your computer connected to the internet, use the web version of the installer.



MySQL Community Server 5.6.11

Looking for previous GA versions?

Select Platform:

Microsoft Windows

Recommended Download:

MySQL Installer 5.6 for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.

I have just installed this *on a windows 7 system* and these are some of my notes. The installer wanted to include applications that work with Excel and Visual Studio- which I do not have on this laptop. If you accept the default Developer system and do not have Excel and/or Visual Studio, then select the Custom install and unselect those two items- we do not use them in class.

I did not install the client Workbench because I do not like it. It is ok if you want to try it.

I accepted defaults for everything else.

When the installation was complete, I had an item that runs the MySQL command line client. **That opens with a request for the password and it is running as root.** You can also run the mysql command line client from a command prompt. You will need to do this in order to run the mysl batch command for the assignments. In order to do this, you need to use a path to the mysql.exe file.

If I go to a regular command prompt and enter

```
   mysql -u root -p
```

I get an error message that the program cannot be found. I have two choices.

(1) Use the full path to the mysql executable file. On my system that would be a long path- I have used \. . . .\ in the example below to hide part of my path since your path might not be the same. You cannot just copy and paste this. Note the quotes since the path includes spaces- and this is a Windows path.

```
   "C:\Program Files\Mysql\. . . .\mysql.exe"  -u root -p
```

(2)Set the PATH environment variable; then you can use just the command mysql and your OS will know where to look for that file. Check the details for doing this with your system. Be certain that you edit the path by adding a semicolon and this new path to the PATH variable- do not replace the current path with the path to mysql or other programs on your system will not start up as you want. On my Windows 7 system, I had to log off and on for the change in the Path variable to be effective.

This is a link to the MS(windows) page for setting your path.

http://msdn.microsoft.com/en-us/library/ee537574%28v=office.14%29.aspx

2. Clients for MySQL

You will also need clients for working with the database. The client software is used to communicate with the MySQL server software. The server software- the dbms- does things such as interact with the tables; lets you create tables and routines. The client software lets the user interact with the dbms. One thing to be concerned with is that different clients have different client commands. For example the \G command work in the mysql command line client, but not in some Gui clients. The default format for numbers, dates, the null value may different with different clients.

For the discussion and demos I use the mysql command line client that is installed with MySQL. You can use this command in interactive mode where you enter a query and get a result, then enter another query and get a result. You can also use the mysql command line in batch mode where the queries are read from a script and execute one after the other. This will be used for assignments and will be discussed in this unit.

When you turn in assignments, I should not be able to tell which clients you used to **develop** your work but you need to use the mysql command line to run the assignment scripts.

2.1. Graphic clients

You can do all of your work in the command line client. Some people prefer to work in a graphical interface. Some of the advantages of using a Gui client to develop your work include

- being able to edit your code in place
- having some version of an Object browser which shows you the names of various database objects such as tables and their columns
- You may be able to generate simple queries by using menus. Code generators tend to add extra parentheses and words such as INNER and ASC when they are not required. You have an obligation to turn in properly formatted code; using a code generator as an excuse for code that does not meet the class standards is not an excuse.

The MySQL folk offer MySQL Workbench as their GUI client. This may have been downloaded when you downloaded MySQL.

The company from Quest Software (http://www.quest.com/toad-for-mysql/) offers a free Gui client Toad for MySQL which runs on Windows.

I do not use either of these clients and do not provide support for them.

3. User accounts and databases

When you install MySQL you will have a root account and password-which you created during installation. You would use this account for administrative purposes only. This class does not deal with administrative tasks. For the things you do in this class, you should create a regular user account that you will use routinely. It is not a good idea to use an administrative account for this class. If you are working on a local computer and do not consider any of your database to be confidential, then working as root is not a major problem for the tasks I ask you to do.

You could use the admin account to create the databases for this class and then create a user with privileges on those databases. Create the database a_testbed and the database a_vets. I am going to use the lower case version of the database name since the case of the database name and the table names can be an issue with some operating systems. If you want to make all the databases for class now, make the following databases. The names of these databases all start with the characters 'a_' as an organization convention (databases for my 155A class start with 'a_' and databases for my 155P class start with 'p_'.

- `a_testbed` we will use this for miscellaneous tables for experimenting
- `a_vets` we use this for the first few demos and assignments
- `a_emp` this and the next two databases are used for most of the demos
- `a_oe`
- `a_prd`
- `a_bkinfo` this and the following are used for later assignments
- `a_bkorders`
- `a_plants` This is used for the midterm and final exams
- `a_xml` this is used at the end of the semester for the xml unit

Demo 01: The command to create a database is `CREATE DATABASE` followed by the database name. You use a semicolon to tell the dbms to execute the command. You would enter this command from within the interactive mysql client (see document 01-03).

```
   Create database a_testbed;
```

A dba would need to take more things into considerations- but this version is sufficient for our databases.

Demo 02: This is the command to create a user named cls_demo with a password of demo_pss. Note the use of quotes in the command. You can use a different id and password.

```
   Create user 'cls_demo'@'localhost' identified by 'demo_pss';
```

You can also create a user who does not need a password by omitting the "identified by" clause. A user without a password would not be a good idea on a production system, but it can be OK for the databases we use in class. The @'localhost' phrase assumes that your installation of MySQL is on your local computer. Pay attention to the quotation marks in these commands.

Demo 03: The command to let that user work with the a_testbed database.

```
   Grant all privileges on a_testbed.* to 'cls_demo'@'localhost' ;
```

Compare this to the grant statement the van Lans uses. First, he grants all privileges to the user BOOKSQL on all of the databases and all of their objects (On *.*); my command limits this user to one specific database. Second, the author includes the clause "With Grant Option" which lets this user grant privileges to other users. When you create a general purpose user for the class, you could use the author's grant syntax so that your general purpose user has full privileges. I am going to be a more cautious administrator and give the user the rights to work with a specific database. The root account can add additional database privileges to a user later.

You can create all the databases first and then grant privileges on those databases to the user you created. I will supply sql for the various tables in these databases as the semester progresses.

4. File names, Extensions, Paths

I expect people taking this class to have sufficient experience with their computer system to handle the following.

The files we will use have a name which consists of a base name and an extension. For example we will have a file named A01_Jones.sql and another file named A01_Jones.lst. These files differ only in the file name extension (sql and lst are file name extensions.)

Some systems try to hide the file name extensions. Some systems say you do not need file extensions. But for the class assignments you do need to have file names such as A01_Jones.lst and A01_Jones.sql and you need to be able to tell these files apart. So you may want to find out how to do this.

You can do an internet search for something like : Mac file extensions or Windows extensions to get directions for your system.

This is a page I found that shows techniques for windows XP, 7, and 8 (and it has pictures).

http://www.bleepingcomputer.com/tutorials/how-to-show-file-extensions-in-windows/

File path- this is a string that says where to find a file on your file system. The full path starts at the root of the disk- or other storage device- and lists the various folders you need to go through to get to the file. You use the backslash(\) to separate the components of a path. C: represents the C hard drive. (this is the windows version Mac people do a search for Mac file path).

For example

```
C:\db_scripts\A01_jones.sql
```

The next path which is a single string may be to be quoted because it contains spaces.

```
C:\Documents and Settings\Rose Endres\My
Documents\CS\2014_spring\F_150A\5_scripts_150A\150A_vets\vets_creates_150A.sql
```

There is a place in a command line where you have to write the pathname to a file- which of these would you prefer to write? Keep your script files in a folder with a short pathname.