**Installing and Using MySQL**

**Introduction**

This semester, we'll be using an open-source database management system called MySQL for some of our work. MySQL has a long history and has been through many versions; for our work, you'll need the latest major version — MySQL 5.0 — which supports some fairly important database features, such as views and stored procedures, that previous versions of MySQL did not.

This document explains how to install and use MySQL on your own computer.

**What is MySQL?**

MySQL is an open-source database management system, which is available on a variety of platforms, including Windows, Mac OS X, and various distributions of Linux and Unix. It is centered around MySQL Server, which is the component that manages data and executes SQL statements that are issued to it. MySQL Server provides services to other programs, such as stand-alone applications with graphical user interfaces, web servers, and application servers.

In order to use MySQL Server, it's necessary to have a separate program that connects to it. For our use, since we want to issue SQL statements directly to the server, a good choice is the MySQL Monitor, also known as **mysql**, which is described in the document below. For those of you who prefer a graphical client instead, you can try the [MySQL Query Browser](http://dev.mysql.com/downloads/query-browser/1.1.html).

**Downloading MySQL**

The latest version of MySQL can be downloaded from <http://dev.mysql.com/downloads/mysql/5.0.html>; it is available free of charge for non-commercial use, including use in our course. You'll no doubt notice that there is a huge number of builds to choose from, supporting Windows, Mac OS X, various distributions of Linux and Unix, as well as other platforms. The latest version, as of this writing, is 5.0.22; new versions arrive fairly frequently, so there could be a newer version available by the time you read this. If so, download the latest one, though if you've already got a 5.0 installation, you won't need the latest minor release for this course. (There is a MySQL 5.1 available, as well, but it's still in an early testing phase, and isn't necessarily stable enough for our use. Since we won't be depending on any of its new features, it would be best for you to stick with 5.0.)

If you want to download MySQL for Windows, there are three choices listed on the [download page](http://dev.mysql.com/downloads/mysql/5.0.html) that I linked above. I suggest downloading the choice marked "Windows (x86)", which is the complete version of MySQL 5.0, along with an installation program that will make it easy to install. This is a fairly large download, weighing in at over 30MB, but be patient and wait for it to finish; after it's complete, proceed to the next section of this document, which assumes that you've downloaded the "Windows (x86)" version of MySQL 5.0 and intend to install it on Windows.

(For those of you running other operating systems, you're on your own to choose and install the version of the software that will work on your computer.)

**Installing MySQL (on Windows)**

The "Windows (x86)" version of MySQL 5.0 is provided in a Zip archive, which contains one file called "Setup.exe". Unzip the archive and run "Setup.exe".

When the setup program begins, it will ask you whether you want a "Typical", "Complete", or "Custom" installation. I suggest choosing a "Typical" installation, since that suffices for our course, and will save you from having to answer questions that you may not be prepared to answer.

Next, you'll be asked to sign up for a MySQL.com account. This is up to you, and it's not a requirement.

After that, you'll be told that the setup wizard is finished, and you'll be offered the chance to "Configure the MySQL Server now". Make sure this checkbox is checked (it should be by default) and click "Finish".

You'll now be taken into a new wizard that will configure your installation of MySQL. What follows is a list of choices that I suggest you make as you work your way through each page of the wizard:

1. Select "Standard Configuration" and click "Next".
2. Make sure the checkboxes "Install As Windows Service", "Launch the MySQL Server automatically", and "Include Bin Directory in Windows PATH" are all checked, and use "MySQL" as the service name. Click "Next".
3. Make sure the "Modify Security Settings" checkbox is checked and that the others are not. Choose a password and enter it into the "New root password" and "Confirm" boxes. *Don't forget this password!* Click "Next".
4. Click "Execute" to start the configuration process. When it's done, click "Finish".

Congratulations! MySQL 5.0 is now installed on your computer and ready to use for your assignments (or whatever else). The "server" will be running in the background whenever your computer is running, so you can connect to it anytime using the **mysql** command-line program (or the graphical MySQL Query Browser tool, if you prefer).

It's possible from time to time that you may need to know what port MySQL Server is running on. By default, it runs on port 3306.

**Using the mysql command-line tool**

If you're running Windows and you followed the installation instructions above, MySQL Server will be running in the background at all times, whenever your computer is on. (If you're running MySQL on your own computer running something other than Windows, be sure you've set up MySQL so that MySQL Server is running in the background, and you should be able to proceed with this set of instructions, as well.)

Start up a "command prompt." You can then connect to MySQL Server using the following command:

mysql --user=root --password

The command-line parameters tell **mysql** to connect to the server using the username "root" (which was created for you during the installation process and has full rights to do anything) and to allow you to type the root password (the password for the "root" username) when the program starts. Without the --password option, you simply won't be able to connect to the server, so be sure to include this option.

When asked, type the root password you specified during installation, and you should see something like this:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 5 to server version: 5.0.18-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

The prompt **mysql>** is where you type SQL commands. To verify that things are working correctly, go through the following steps to create a simple database and query it. In the text below, boldfaced text indicates something that you should type, while normal-weight text indicates something that the **mysql** program will print in response.

mysql> **CREATE DATABASE test123;**

Query OK, 1 row affected (0.00 sec)

mysql> **USE test123;**

Database changed

mysql> **CREATE TABLE customer(**

->  **customer\_id INTEGER,**

->  **customer\_name CHAR(30),**

->  **customer\_city CHAR(20),**

->  **PRIMARY KEY(customer\_id));**

Query OK, 0 rows affected (0.16 sec)

mysql> **INSERT INTO customer VALUES (1, 'Ann', 'Taipei');**

Query OK, 1 row affected (0.08 sec)

mysql> **INSERT INTO customer VALUES (2, 'Joe', 'Chiayi');**

Query OK, 1 row affected (0.09 sec)

mysql> **SELECT \* FROM customer**

-> **WHERE customer\_city = 'Taipei';**

+-------------+---------------+---------------+

| customer\_id | customer\_name | customer\_city |

+-------------+---------------+---------------+

| 1 | Ann | Taipei |

+-------------+---------------+---------------+

1 row in set (0.00 sec)

mysql> **DROP DATABASE test123;**

Query OK, 1 row affected (0.09 sec)

mysql> **\q**

Bye

There are a few things I want you to notice in the example above:

* The **CREATE DATABASE** and **DROP DATABASE** commands are used, respectively, to create and destroy databases. It's not necessary to destroy a database before logging out of the server, but I thought it would be nice to include this in the short example here, so that there would be nothing left over from it when you were finished with it.
* MySQL Server can simultaneously manage multiple databases. Before you can issue SQL commands related to some particular database, you'll have to tell **mysql** which one you want to use. This is done with the **USE** command, as above.
* Commands do not have to be typed on one line. If you want to spread them out over several lines, pressing Enter at the end of one line of the command will cause **mysql** to ask you for the next line. It will continue asking you for more lines of input until the command is complete. When the command is complete, it's executed.
* The **mysql** command-line tool expects all SQL commands to end in a semicolon. This is necessary because it isn't otherwise clear when a complete command has been typed. (Semicolons, in this context, are used to overcome the user interface limitation of typing SQL commands in a command-line interface; other tools or programs connecting to MySQL Server do not need to include these semicolons and, in general, semicolons are not required at the end of SQL statements.)
* The result of most commands includes an indication of how long they took to execute. Your times likely won't match the ones that I got when I ran the example above, but this difference is irrelevant for our purposes.
* The "\q" command exits **mysql** and takes you back to the command prompt.