Installing and Using XAMPP on Windows

March 9, 2020

This is what I used when I taught CSE 305 (Database Systems) in 2020. I think the info on this page is still valid. If necessary, you may only need to adjust slightly, I think.
This page contains information that you will find useful as you install XAMPP on your Windows machine and use it with JDBC.

1. Installing XAMPP on Windows

Follow these steps:

1. Go to https://www.apachefriends.org/index.html.

2. Download **"XAMPP for Windows 7.4.3 (PHP 7.4.3)"** (or the latest version) into a folder of your choice. Adjust in the rest of this page if you are using a different version. You will download xampp-windows-x64-7.4.3-0-VC15-installer.exe.

3. Execute the installer. As you go through the installation process, you will find this page quite helpful.

   If you continue with the **XAMPP Control Panel** toward the end of the installation, you will get to try the **shell command-line utility**. Use the following commands:

```
     # mysql -h localhost -u root    -- To start a client from
shell.

     MariaDB [(none)]> show databases;  -- One possible command.
     /*
     I tried the above statement/command to see if it worked or
not.
     When you start MySQL, you will actually get MariaDB instead.
     MariaDB (not MySQL) is included in XAMPP, but MariaDB is
similar
     to MySQL in its interface.  In fact, it is created by the
     ex-MySQL folks, I read.
     */
```

Now, continue with section 2 below.

2. Using a MySQL (MariaDB) Client

In the install folder you will see the following three programs:

- xampp_start.exe
- xampp_stop.exe

- xampp_control.exe

Use these to **start** and **stop** the server and for the **control panel**. Try **not** to start one when one is already running. It would not hurt to try to stop it when it is not running. That would be one way to avoid trying to start another one when one is already running.

Once you start the **Control Panel**, start MySQL and try the **shell** associated with it. That is one way to run a MySQL (well, MariaDB) client. Once you start the shell, try the usual SQL statements. You will see some of them in the next section.

3. Trying SQL with MariaDB

Once you start a MariaDB client in a shell, you can try some database commands and SQL statements. Below are some examples:

- To see a listing of all the currently available databases, try:

  ```
  MariaDB> show databases;           // don't forget the ';'
  ```

- You can create a new database named artlee like this:

  ```
  MariaDB> create database artlee;
  ```

- You can create a user while granting privileges to that user. To grant a user access to the database from the localhost, use the command:

  ```
  MariaDB> grant all privileges on database.* to user@localhost identified by 'password';  // don't forget the single quotes.
  ```

  where database is the name of a database, user is the name of a new user, and password is a password. For example, I used artlee, art, and lee respectively.

- To grant the user rights to the database from other client machines, use the command:

  ```
  MariaDB> grant all privileges on database.* to user@"%" identified by 'password';
  ```

  where @"%" acts as a wildcard for access to the database from any client machine. Here again database is the name of a database, user is the name of a user, and password is a password. For example, I used artlee, art, and lee respectively.

- If you want some help, try:

  ```
  MariaDB> help;
  ```

- To select a database (e.g., named 'artlee'), which you will need to do before you start using it among all the databases that you have created so far, try:

  ```
  MariaDB> use artlee;
  ```

- To delete a database named 'artlee', try:

  ```
  MariaDB> drop database if exists artlee;
  ```

- Running a script file. You may create a database and populate the database with some tables and records in the tables. It is more convenient to do it using a script file. If you want to run the script file: db.sql located for example in c:/alee/cse305/db/db.sql, try:

```
MariaDB> source c:/alee/cse305/db/db.sql;
```

Take a look at the content of the script file db.sql. By the way, how would you create a script file? Use your favorite text editor.

- If you want to leave the server, try:
  ```
  MariaDB> quit;
  or
  MariaDB> exit;
  ```

4. Some More Examples of SQL Queries

- Delete the database named artlee first if it exists so that you know you are starting with a clean state:
  ```
  mysql> drop database if exists artlee;
  ```
- Now, create a new database named artlee:
  ```
  mysql> create database artlee;
  ```
- To select a database named artlee:
  ```
  MariaDB> use artlee;
  ```
- Creating a table in a database (artlee in this case)
  ```
  MariaDB> create table Accounts (ID INTEGER, Balance INTEGER);
  ```
- See the contents of the table just created (of course, it should be empty at this point):
  ```
  MariaDB> SELECT * FROM Accounts;
  ```
  Use this to see if insert worked or not when you try inserts below.

- Creating a relation in a table (three times)
  ```
  MariaDB> insert into Accounts VALUES (1, 11);
  MariaDB> insert into Accounts VALUES (2, 22);
  MariaDB> insert into Accounts VALUES (3, 33);
  ```
- Add up the balances in the Accounts relation:
  ```
  MariaDB> SELECT Sum(Balance) FROM Accounts;
  ```
- Add another table and some tuples into the table (I intentionally used lower case letters here to show that it is case insensitive, but try to use a good style in naming):
  ```
  MariaDB> create table Names (id integer, dob date);
  MariaDB> insert into Names values (1, "2020-03-09");
  MariaDB> insert into Names values (2, "2018-03-20");
  MariaDB> insert into Names values (3, "2017-01-20");
  ```

- The cartesian product of the Names and Accounts relations if both exist:

  ```
  MariaDB> SELECT * FROM Names, Accounts;          -- (+)
  ```

  Use '--' to add a comment.

- Keep only those records from (+) that agree on ID

  ```
        MariaDB> SELECT * FROM Names, Accounts WHERE Names.ID =
  Accounts.ID;
  ```

- Keep only those records from (+) that have a DOB prior to 1/1/2018:

  ```
        MariaDB> SELECT * FROM Names, Accounts WHERE Names.ID =
  Accounts.ID
               AND DOB < Date("2018-01-01");
  ```

- Keep only the Names.id, Balance, and DOB columns:

  ```
        MariaDB> SELECT Names.id, Balance, DOB from Names, Accounts
  WHERE
               Names.ID = Accounts.ID AND DOB < Date("2018-01-01");
  ```

- Try more that you can think of. Find a reference if you want to learn more about these.

5. Using MySQL with Java (JDBC)

Note: This section is written by me, but I have not actually tried it on a machine, but the TAs tried it on their machines and worked for them.
In this section, I describe what you would need to be able to use MariaDB from Java using JDBC. I assume that you have MariaDB (XAMPP) installed.
Find a Java connector, MariaDB Connector/J 2.5 (the latest at this writing) from here. Click on "Download 2.5.4 Stable Now" followed by "MariaDB Connector/J .jar files" ("Universal"). Then, click on "mariadb-java-client-2.5.4.jar". Download it to a folder of your choice. (Pick a folder that will be around since you will use the complete path name of the file below). This is the connector that you will need. The connector enables Java to talk to the database server.
You will need to add that jar file using its complete pathname to the CLASSPATH environment variable. This is how you do it. Try the following if you are using a Windows 10 machine. For other versions of Windows, find a way that works:

```
    Start (the four square thing at the lower-left corner)
 -> Control Panel
 -> System and Security
 -> System
 -> Advanced System Settings (on the left column)
 -> Environment Variables
```

You will see **User variables** and **System variables**. Under **System variables** you may or may not see CLASSPATH. Select it if exists and edit to add that jar file using its complete pathname to the end of the value of CLASSPATH preceded by a semicolon. If you don't see CLASSPATH, then use "New..." and add CLASSPATH as "Variable name" and the complete path of the jar file as "Variable value".

Now, download, compile, and run Bank1.java which will bring in DB.java. They use JDBC. Be sure to read the comments at the top of Bank1.java to see how to run it.

- jdbc.sql (Run this script from a command line prompt to set up the database first. Read the file. It creates a simple bank database.)
- DB.java
- Bank1.java (Run this one)

and read the instructions at the top of Bank1.java to run and test your JDBC capability. Also read DB.java to see how a database driver is loaded and how a connection is established.

If This Does NOT Work and nothing else could be tried to make it work, uninstall XAMPP and MariaDB and then follow the instructions *very carefully* from the very beginning again. Some students had to do that and it worked for them.
That's it!