

Hydroinformatics

Assignment 4 Instructions

Prerequisites

To complete this exercise on your own computer, you must first install MySQL, MySQL Workbench, and the ODM Tools Python Software. These software tools can be downloaded and installed using the instructions provided on the Canvas site.

Introduction

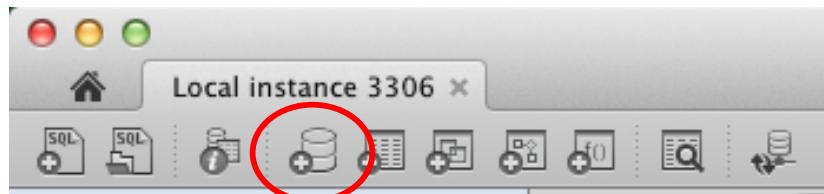
We will begin to populate your ODM database with information about the sites at which data have been collected, the variables that have been measured, the methods that have been used, etc. One of the easiest ways to do this is to organize the data into a set of comma separated values (CSV) data files to be uploaded to the database.

To use the data loading tools available with MySQL Workbench, the CSV data tables to be loaded must be essentially identical to the tables within the ODM database into which you are loading them. First, however, we need a blank ODM database into which we will load the data.

NOTE: Organizing your data so that it can be loaded into an ODM database involves assembling all of the necessary information to populate the metadata within the database and then formatting it into a set of CSV files that can be loaded into the database. For the first part of this assignment, we have done some of this work for you. However, you will have to load data for an additional data collection site that you select. You can look at the format and contents of the files that we have created to get an idea of what you need to do for the second part of the assignment.

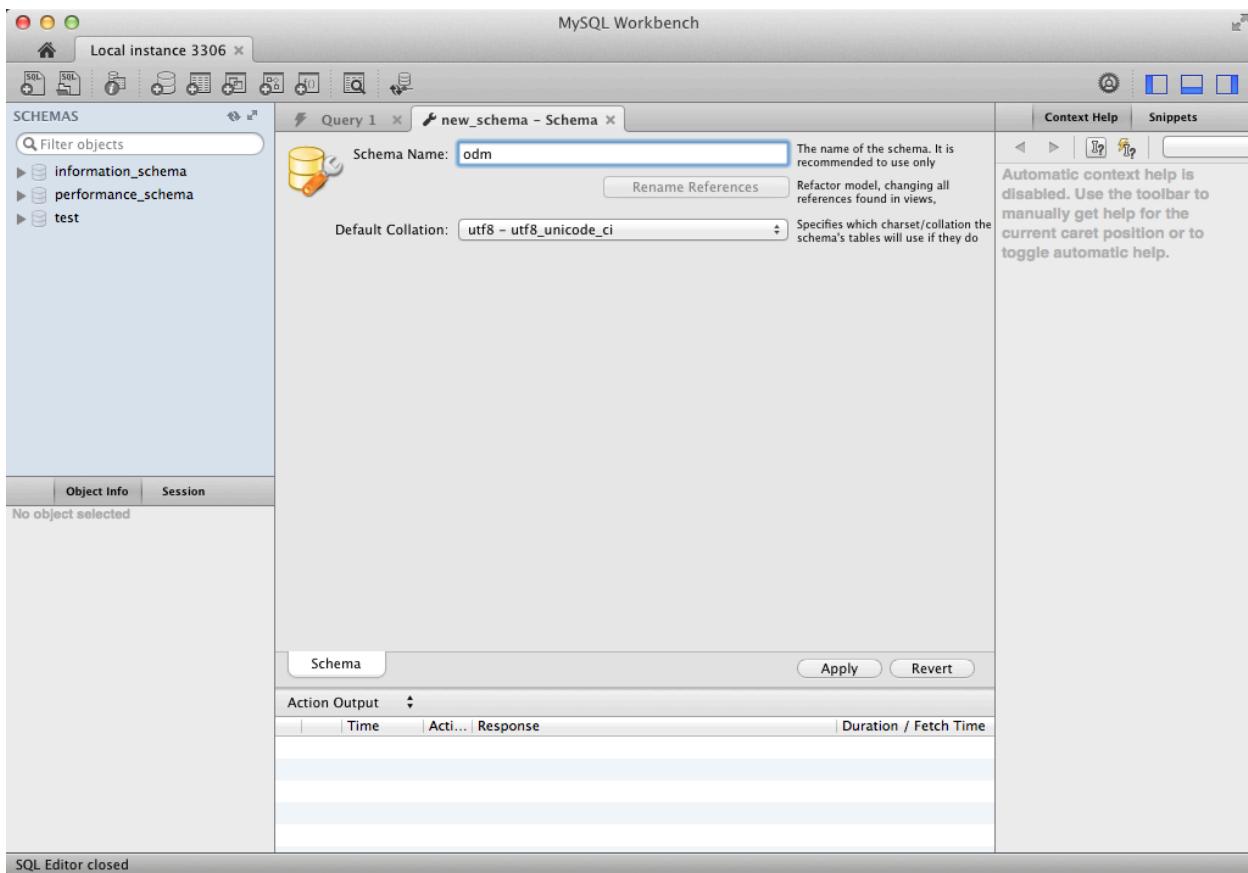
Creating a Blank ODM Database

1. Download the SQL script from the Canvas website (ODMforMySQL_1.1.1.sql).
2. Open MySQL Workbench and click on your local MySQL database server connection.
3. Create a new database schema by clicking on the “New Schema” button on the toolbar at the top of the window:

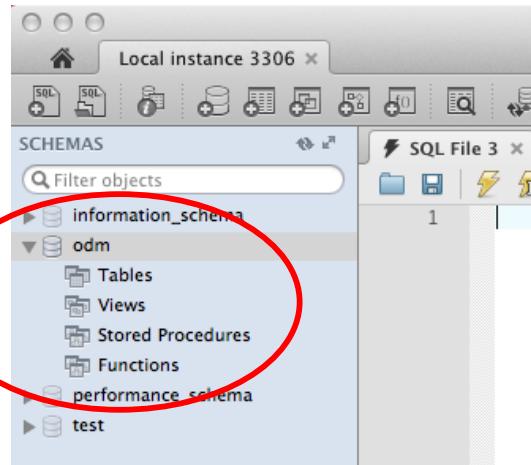


NOTE: In MySQL, there is no difference between a “database” and a “schema”. So these words can be used interchangeably.

4. Name your schema name to “odm”, and set the Default Collation to “utf8 – utf8_unicode_ci”, then click the “Apply” button.



5. A window will pop up with the SQL code needed to perform the action you just completed. Click the “Apply” button.
6. A window indicating that your SQL statement successfully completed should now be showing. Click the “Close” button.
7. You will now notice that there is a new database in your list of schemas. Click the arrow next to the “odm” schema you just created to expand the view. You will see “Tables”, “Views”, “Stored Procedures”, and “Functions”. This is where you can access the objects in your database. But, there aren’t any yet because your database is blank, so let’s create those now.



8. On the MySQL Workbench File drop down menu select “Open SQL Script”. Using the file browser, select the “ODMforMySQL_1.1.1.sql” script that you downloaded from the Canvas website. The script will open in a new tab in MySQL Workbench.

```

1 -- MySQL Dump Script for Creating an ODM 1.1.1 Blank Schema with
2 -- Created by: Jeff Horsburgh, Utah State University
3 -- Modified by: Jiri Kadlec, Aalto University (changed table name)
4 -- Created on: 1-3-2012
5 -- Modified on: 04-29-2013
6 -- Note: Check to make sure the collation of your database schema
7
8 -- Ensure that tables with existing primary key values of zero are
9 SET sql_mode='NO_AUTO_VALUE_ON_ZERO';
10
11
12 -- Table structure for table `generalcategorycv`
13
14
15 • CREATE TABLE `generalcategorycv` (
16   `Term` VARCHAR(255) NOT NULL,
17   `Definition` TEXT NULL,
18   PRIMARY KEY (`Term` ASC)
19 ) DEFAULT CHARSET=utf8 collate utf8_unicode_ci ENGINE=InnoDB;
20
21
22 -- Table structure for table `samplemediumcv`
23
24
25 • CREATE TABLE `samplemediumcv` (
26   `Term` VARCHAR(255) NOT NULL,
27   `Definition` TEXT NULL,
28   PRIMARY KEY (`Term` ASC)
29 )
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

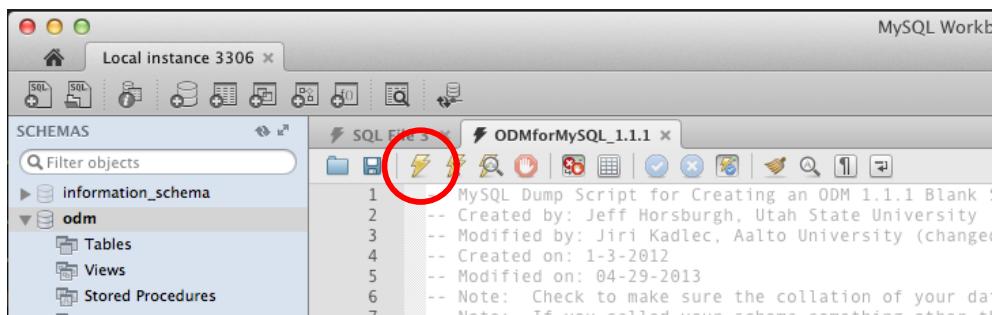
```

Action Output

	Time	Action	Response	Duration / Fetch Time
1	14:26:07	App...	Changes applied	

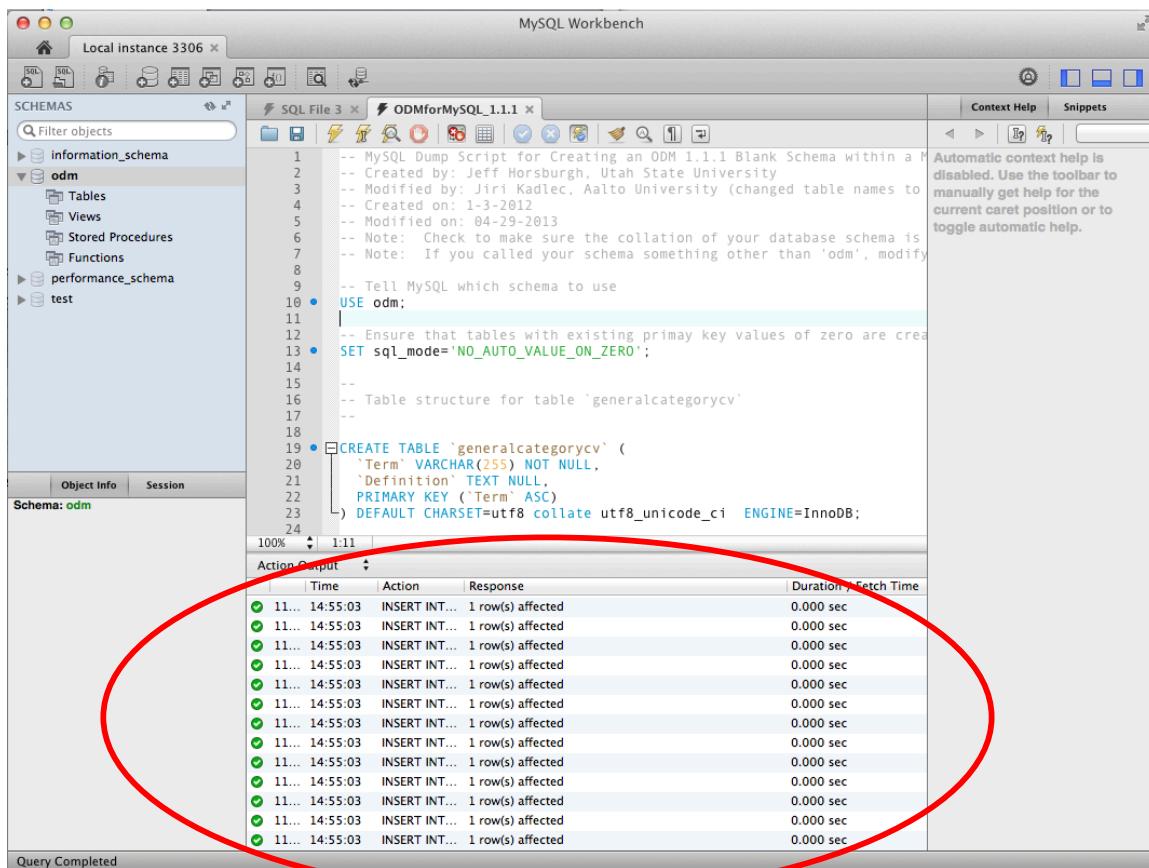
Loaded SQL script file '/Users/JeffHorsburgh/Documents/Working/Teaching/CEE6110 2014/Lectures/Class 10 - Database Implementation/ODMforMySQL_1.1.1.sql'

9. Click the run button (looks like a lightning bolt) on the toolbar to execute the script.

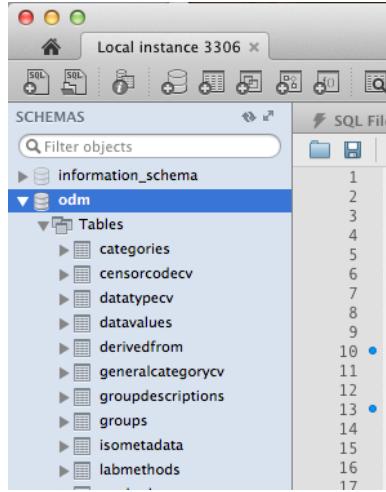


NOTE: This SQL script was created to automatically build all of the tables, attributes, relationships, etc. for ODM within a MySQL database. It is the standard distribution of ODM for MySQL.

10. You will notice that messages are printed to the Action Output pane as MySQL Workbench executes the script.



11. When the script is finished running (you should see a message that says "Query Completed" in the bar at the very bottom of the MySQL Workbench window), right click on the "odm" schema in the list of schemas in the right panel and select "Refresh All". Now if you click the arrow next to the "odm" schema, you should be able to expand the tables section and see all of the tables that have been added the database.

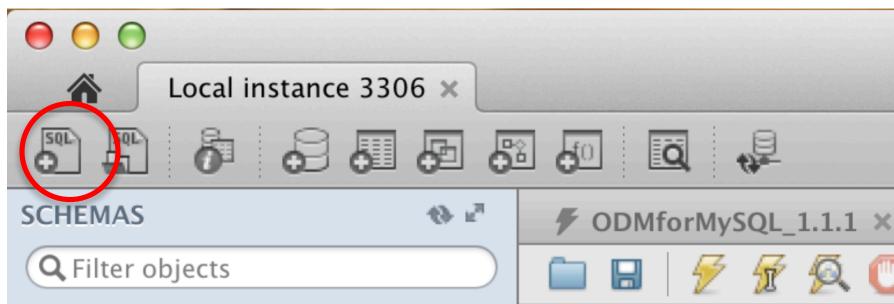


12. That's it – you now have a blank ODM database in MySQL that you are ready to load data into.

Load Table Based Data into the ODM Database in MySQL

For the next part of this exercise, a set of data files containing data for the Logan River GAMUT Network sites has been prepared for you. Use the following steps to begin loading data into your ODM database:

1. Download the zip file containing the data files for this Assignment from the course Canvas website. Extract the files to a folder on your desktop.
2. Open MySQL Workbench.
3. Click on the “New SQL Tab” button on the toolbar at the top of the MySQL Workbench window.



4. A new SQL tab will open for you to create a new script. We are going to use the “[LOAD DATA LOCAL INFILE](#)” capabilities of MySQL to load the data for this exercise. In your new script tab, type the following lines of code.

```
LOAD DATA LOCAL INFILE '/Users/JeffHorsburgh/Desktop/Assignment4Data/sites.csv'
INTO TABLE sites
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
```

```
(SiteCode,SiteName,Latitude,Longitude,LatLongDatumID,Elevation_m,VerticalDatum,LocalX,  
LocalY,LocalProjectionID,PosAccuracy_m,State,County,Comments,SiteType);
```

The first line specifies which CSV file to load using the full path to the file on your computer. The second line specifies which database table you are loading data into (in this case “sites”), the third line gives the column delimiter in the file (in this case a comma), the fourth line gives the text qualifier for text fields in the file (in this case a double quotation mark), the fifth line gives the end of line delimiter character in the file (in this case both carriage return and new line characters), the sixth line tells MySQL to ignore the first line of the CSV file (which contains the column headers), and the last lines tells MySQL which columns/fields within the sites table in the database that data are being loaded into and in which order.

NOTE: In the first line of this code, you will have to change the file path to wherever you unzipped the data files on your computer. Also note that we are not loading the SiteID column to the sites table in the database. This field is an auto-increment field that is automatically populated by MySQL when you insert new records.

NOTE: When loading data using CSV files, you MUST supply values for every required field in the table. You can choose to omit the optional fields if you want, but in the last line of the code above, you should only list the fields that you are actually loading from the file.

5. Execute the above script and then right click on the sites table and choose “Select Rows”. This will show you a tabular view of the records that you just loaded into the sites table from your sites.csv file.
6. Follow the steps above to load the “methods.csv” table, the “sources.csv” table, and the “variables.csv” table. Don’t forget to change the path to the file you are loading, the name of the table you are loading data into, and the list of fields you are providing in the files.

NOTE: Now that you have added data to the sites, methods, sources, and variables tables in your ODM Database, you can use MySQL Workbench to explore the contents of those tables. Although we have only loaded sites, variables, methods, and sources, you can load data into any of the tables in your ODM database using the method shown above, including datavalues.

Load DataValues into the ODM Database

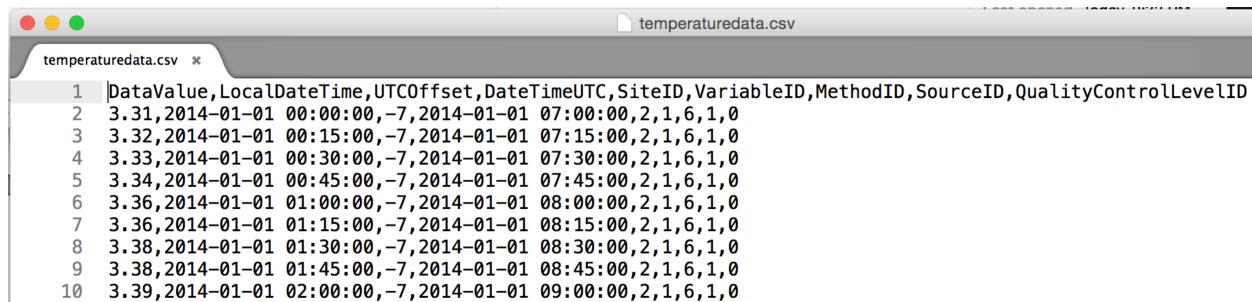
In this section, we will load some in situ sensor data collected at one of the sites in the Logan River into your ODM database. I have given you an actual data file from one of the monitoring sites in the Logan River. But, it’s not in the right format to load into the ODM datavalues table, so you are going to have to do some reformatting.

It’s easiest to do this one data column at a time. We’ll start with the “WaterTemp_EXO” column, which contains the water temperature measured in degrees C. Since the ODM DataValues table uses the foreign key IDs for the Site, Variable, Method, Source, and QualityControlLevel information, you need to review those tables in the ODM database and match the descriptions above to the correct SiteID, VariableID, SourceID, MethodID, and QualityControlLevelID. Then, you need to create a CSV file in the right format to load into the DataValues table. Keep in mind that these are raw data, so the QualityControlLevelID will be 0.

For the “WaterTemp_EXO” column, I have provided a sample formatted CSV data file that you can load into the database. Here’s a couple of notes about that file:

1. It only contains the fields required by the ODM DataValues table.
2. Note the format of the Date/Time columns. You must format your dates this way in the CSV file if you want MySQL to recognize the dates when you upload the file.
3. If you make your CSV files on Windows, the end of line character will most likely be ‘\r\n’. If you make them on a Mac, you may have to use a different end of line character, and it will most likely be ‘\n’ (see line 5 in the code above).

A snippet of the CSV file to load into the database for the “WaterTemp_EXO” column looks like the following:



The screenshot shows a Mac OS X TextEdit window with a dark theme. The title bar says "temperaturedata.csv". The content area displays 10 rows of data, each starting with a line number (1-10) followed by a vertical bar and then the data values. The data values are comma-separated and include a timestamp, a value, and several IDs.

	DataValue	LocalDateTime	UTCOffset	DateTimeUTC	SiteID	VariableID	MethodID	SourceID	QualityControlLevelID
1	3.31	2014-01-01 00:00:00	-7	2014-01-01 07:00:00	2,1,6,1,0				
2	3.32	2014-01-01 00:15:00	-7	2014-01-01 07:15:00	2,1,6,1,0				
3	3.33	2014-01-01 00:30:00	-7	2014-01-01 07:30:00	2,1,6,1,0				
4	3.34	2014-01-01 00:45:00	-7	2014-01-01 07:45:00	2,1,6,1,0				
5	3.36	2014-01-01 01:00:00	-7	2014-01-01 08:00:00	2,1,6,1,0				
6	3.36	2014-01-01 01:15:00	-7	2014-01-01 08:15:00	2,1,6,1,0				
7	3.38	2014-01-01 01:30:00	-7	2014-01-01 08:30:00	2,1,6,1,0				
8	3.38	2014-01-01 01:45:00	-7	2014-01-01 08:45:00	2,1,6,1,0				
9	3.39	2014-01-01 02:00:00	-7	2014-01-01 09:00:00	2,1,6,1,0				

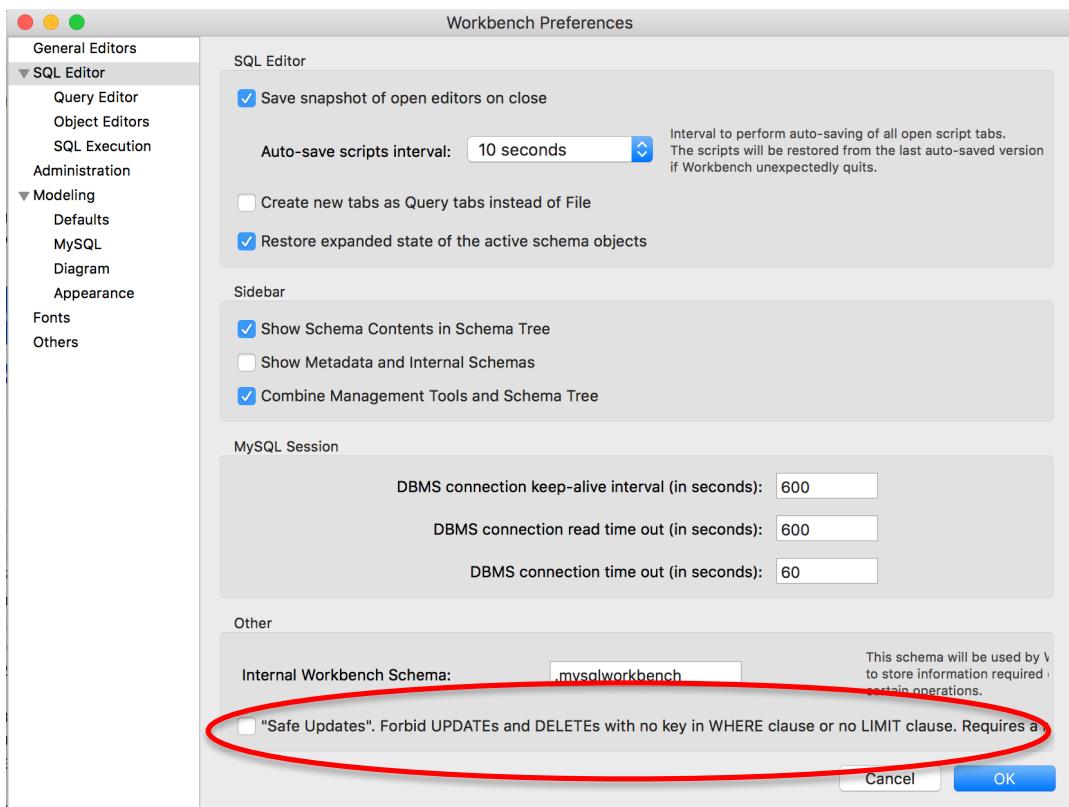
You should be able to load the water temperature data provided (temperaturedata.csv) using the same code listed above, but changing the file name, table name, and field names.

Now, you create some additional DataValues files to load using whatever tools you want to use (Excel is one easy option). Save them as CSV files, and then load them into the database. The details you need to match the metadata in the database with the column in the file are in the header of the data file that I gave you. Remember, that you need to use the foreign key IDs for the correct site, variable, method, source, and qualitycontrollevel in the CSV files that you will load. You don’t have to load all of the columns in the file, but you should do 2 – 3 additional columns.

Update the ODM SeriesCatalog Table

In ODM there is a table called seriescatalog that keeps a listing of all of the time series you have loaded into the database. Once you have loaded the data, the contents of the seriescatalog table can be automatically generated using a stored procedure that is included with the blank ODM database schema. Use the following steps to update the seriescatalog table:

1. First we need to modify an option in MySQL to enable us to run the stored procedure. Open the MySQL Workbench Preferences window (on Mac choose “Preferences” from the “MySQL Workbench” drop down, on Windows choose “Preferences” from the “Edit” pull down menu. The following window will pop up. Choose “SQL Editor” in the tree view on the left and then uncheck the box next to “Safe Updates”.



2. Open a new script window by clicking on the new script button on the toolbar.
3. In the new script window, type the following code:

```
USE odm;
CALL sp_UpdateSeriesCatalog();
```

4. Execute the script by clicking the lightning bolt icon. If you still get an error, you may need to disconnect from your database server and then reconnect in the main MySQL Workbench window (or just close MySQL Workbench and then open it again).
5. Once the script is complete, you should be able to right click on the seriescatalog table and choose "Select Rows."
6. You will now see a tabular view of the series catalog table with a record for each of the time series that you loaded from the datalogger file.

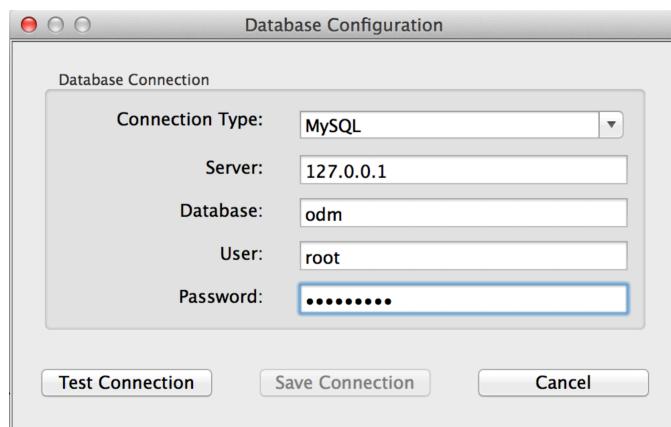
NOTE: The stored procedure sp_UpdateSeriesCatalog is just a SQL query that has been stored in the database for you to execute whenever you need it. If you want to see the code of the stored procedure, expand the Stored Procedures group under the odm database and right click on sp_UpdateSeriesCatalog. Then choose "Alter Stored Procedure" from the menu that pops up. A new script tab will open with the code of the stored procedure. Make sure you don't save any changes to the stored procedure or it will not work correctly. When you are done having a look, just close the script tab.

Visualize and Manage Data in the ODM Database

Now that you have loaded some data into your ODM database, you can use the ODM Tools Python software to do some quick visualizations and manage your data. In the next part of the Assignment we will use ODM Tools to examine the contents of your ODM database.

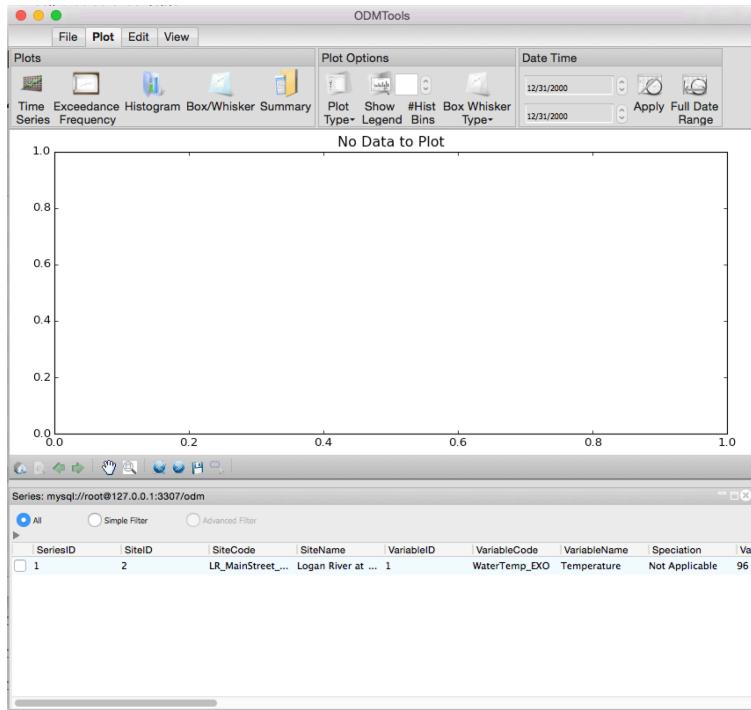
To examine the data you just loaded using ODM Tools:

1. Follow the instructions for downloading the ODM Tools software on the course Canvas website.
2. On Windows, Open ODM Tools by clicking Start → All Programs → CUAHSI HIS → ODM Tools Python. On Mac, select ODM Tools Python from your Applications directory (or wherever you extracted the ODM Tools Python App). The ODM Tools “Database Configuration” window will open. Enter the connection information for your ODM database. When your window looks like the following, click the “Test Connection” button. On the Test Connection window that pops up, click the “OK” button. Then, click the “Save Connection” button.

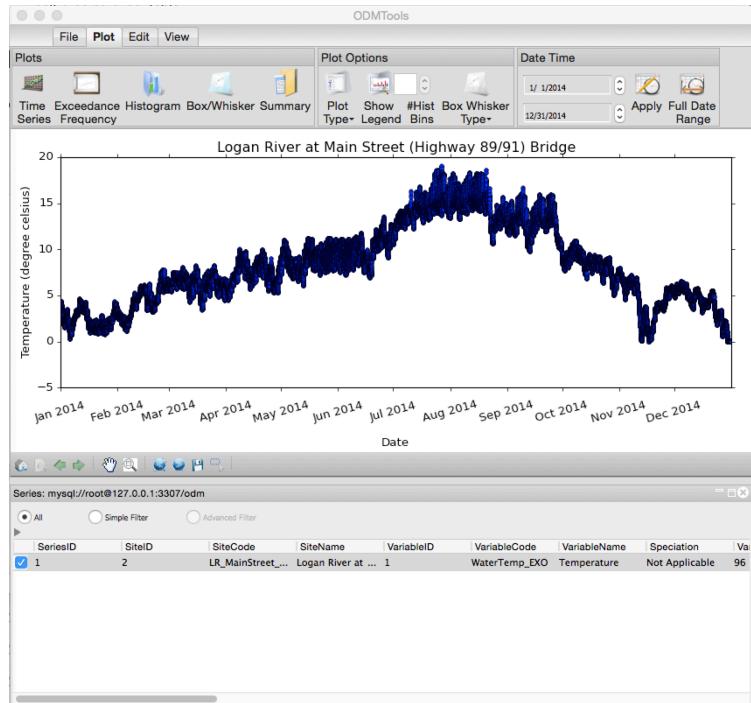


NOTE: If you are a Mac user and you are unable to open the ODM Tools Python App because you get a warning that it is from an untrusted source, press the Control key, then click on the App icon, and choose Open from the shortcut menu. This will save an exception to your security settings and will enable you to open the app in the future just by double clicking it.

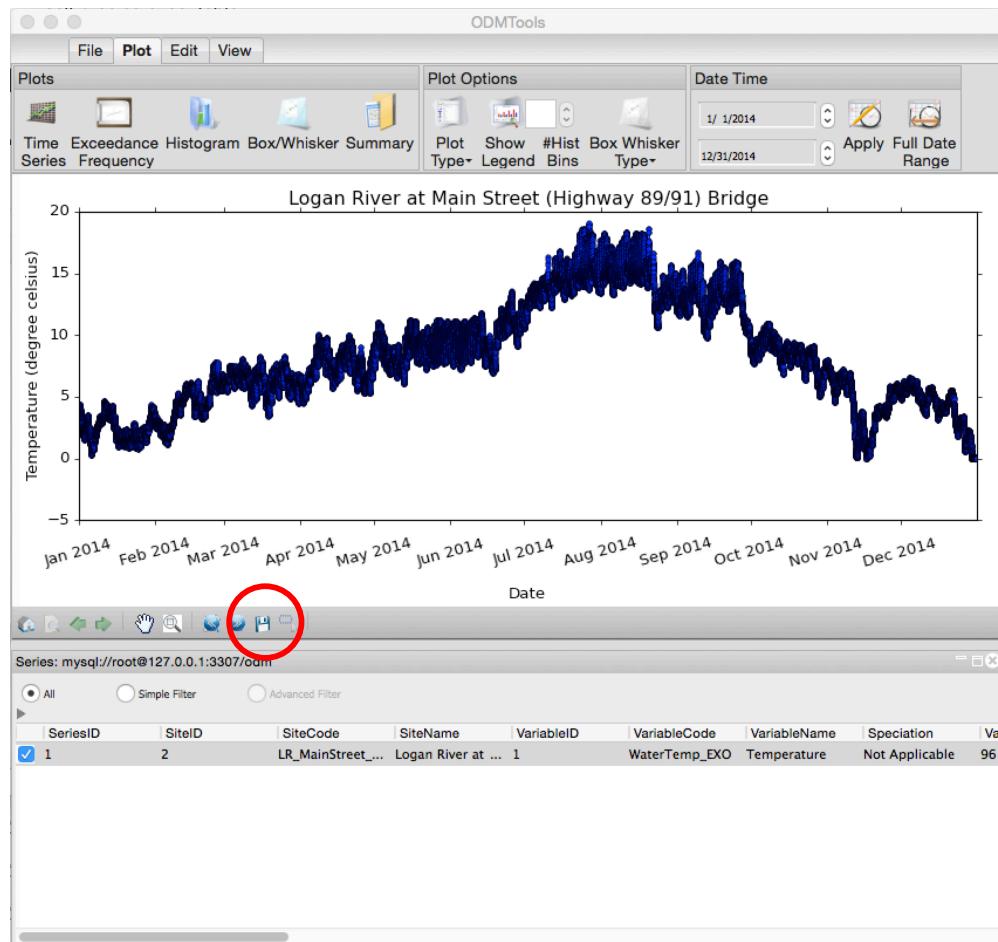
3. The ODM Tools application will now open and you will see something like the following.



4. Since your ODM database only has a small number of data series, you should see all of them in the series selection tool at the bottom of the window. In the series selection tool, you can specify what you would like to be shown in the plot window by clicking on the check boxes next to the series in the list. In the example below, I have only loaded one time series into my database.



- At the top of the form, you will see a ribbon toolbar with options for switching between plot types and other options for the software. You are welcome to explore the functionality of ODM Tools.
- Once you have created a plot that you want to export, click on the save button on the plot toolbar and then use the file browser that pops up to export an image of the plot. This image can then be imported into a document (like your homework!)



Editing Data Using ODM Tools

You may notice as you are visualizing the data that you loaded into your database that there are some “interesting” artifacts. These are raw data from the field and have not been subject to any QA/QC procedures. As an advanced step, you can explore using the data editing functionality of ODM Tools (the Edit tab at the top of the form) to clean these data up a little bit.

NOTE: You may want to make the plots required for your homework before you start fiddling with the edit tools.

1. Click on the "Edit" tab at the top of the ODM Tools form. Select a time series that you want to edit in the series selection tool at the bottom of the form by clicking on the row in the table and making sure the check box next to it is checked so that it is plotted.
2. Then, click on the "Edit Series" button on the toolbar. You will notice that the selected series enters edit mode and turns black. Once in edit mode, any NoData values (e.g., values of -9999) are shown on the plot. You will also notice that a Python script window opens up and the editing buttons on the toolbar become active.
3. Using the buttons on the toolbar, you can select points using the "Filter Points" button. Selected points turn red. You can also select points using the point selection tool on the plot toolbar (it looks like a little blue circle with a white arrow pointing at it).
4. Once points are selected, you can change their value, interpolate them, perform a drift correction, etc. by using the buttons on the edit toolbar.