Connecting to the database through Matlab for Windows

Joseph P Tulenko
ICE-D tutorial Matlab for Windows

February 23, 2022

In this set up tutorial, we will be walking you through how to set up a connection between ICE-D and Matlab on your personal Windows desktop through an ODBC Connection (JDBC might also be an option but we have successfully figured out how to connect via ODBC for now and if it ain't baroque don't fix it).

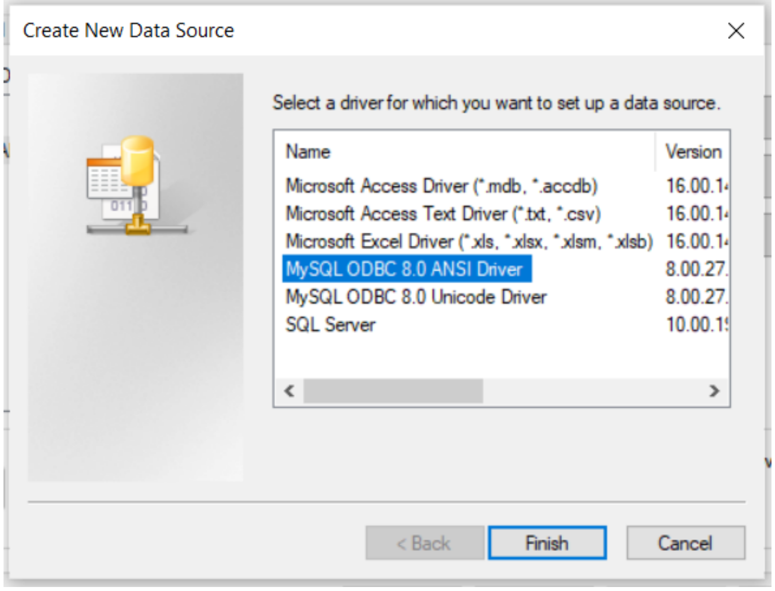
Once you do this, you can format SQL queries directly in Matlab and work with the data without copy+pasting data over from the database into Matlab or excel tables, etc. Neat! See also some of the examples we have come up with so far in the next section on this page, and email us with some of your ideas!

We will be using the Database Explorer APP in Matlab so make sure you have it downloaded. You can do so by navigating over to the APPS tab in Matlab and selecting the "Get More Apps" button. There will appear a new window with a search bar where you can type in Database Explorer and install the App.

Step 1. First, to be able to make an ODBC connection, we need to download a MySQL ODBC connector/driver. You can find a webpage to download the latest driver here (https://downloads.mysql.com/archives/c-odbc/). As of the publication date on this tutorial, I am using version 8.0.27 (64-bit for my new Lenovo Yoga laptop) and it seems to work fine.

Step 2. If you wish to connect to ICE-D using Matlab, the way we currently have it set up is that you actually connect to the local port on your machine that HeidiSQL is using to talk to ICE-D remotely. Therefore, you must have ICE-D open in HeidiSQL while you are using Matlab. Be sure to do that first.

Next, go into the database explorer app, and in the top panel there should be an option to "Configure Data Source". From the dropdown menu, select "Configure ODBC Datasource". I have my connection set up in the "System DSN" so I would recommend navagating over to that tab and then selecting the Add option. The window that appears should look something like this:



Create New Data Source

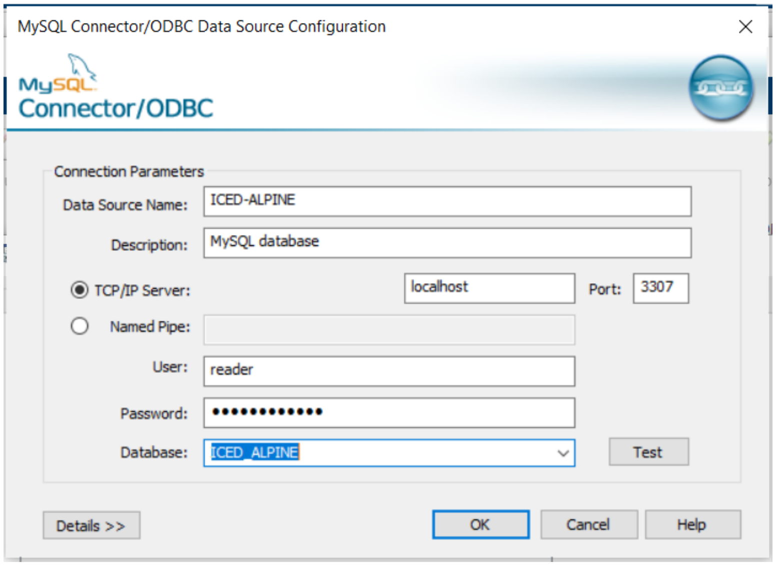
Select a driver for which you want to set up a data source.

Name	Version
Microsoft Access Driver (*.mdb, *.accdb)	16.00.1
Microsoft Access Text Driver (*.txt, *.csv)	16.00.1
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	16.00.1
MySQL ODBC 8.0 ANSI Driver	8.00.27.
MySQL ODBC 8.0 Unicode Driver	8.00.27.
SQL Server	10.00.1

If you successfully downloaded the MySQL ODBC connector/driver, you should see those options in the list on the window above (like the one that I have highlighted in the example above). If you do not see those options this will not work. Go back and check that you properly installed the MySQL ODBC connector/driver.

Select "MySQL ODBC 8.0 ANSI Driver" and hit finish.

Step 3. Now, you should see a screen like this pop up:



MySQL Connector/ODBC Data Source Configuration

MySQL Connector/ODBC

Connection Parameters

Data Source Name: ICED-ALPINE

Description: MySQL database

● TCP/IP Server: localhost Port: 3307

○ Named Pipe:

User: reader

Password: ••••••••••••

Database: ICED_ALPINE

In this example I have filled out all of the necessary information and yours should look similar. It doesn't matter what you c all the Data source name, but I would recommend that it be something logical because you will need to know it later. Same with the description, I just called mine a MySQL database.

For the connection, as stated previously you aren't really connecting directly to the database, but rather connecting to the location/port on your own machine where the SSH tunnel is set up. Thus, the TCP/IP Server location is under localhost (your own machine) and whatever port you specified when setting up the SSH connection - in my case, I chose 3307. Perhaps you chose 3308 or something similar.

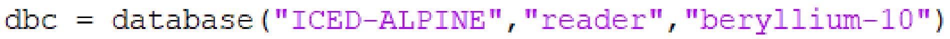
Note: you do not want to try port 3306 because that is the SSH port, it is already in use and as previously stated, you aren't trying to connect directly to the database through Matlab, only your own port that the SSH tunnel is directing information from the database to.

When connecting through Matlab, we do not want any potential mixup with matlab coding where a user may accidentally delete the entire database (yikes!) so we request that you use the ”reader” account only!! Not your own personal account. The reader username and password are as follows:

user: reader
password: beryllium-10

Now, if the final drop down menu titled Database is still blank, have no fear! Try pressing the test button next to the drop down menu. If you have successfully established a connection, then you should get a window that says so, and options should appear in the drop down menu. Recall that you need to have HeidiSQL connected to ICE-D while you are doing this so make sure that is true. Hit OK once everything looks good and you should be all set!

Step 4. For the final step, you should return to the main Matlab command window and type in the following code:



```
dbc = database("ICED-ALPINE","reader","beryllium-10")
```

Note: the first entry in the database command is the name of your database that you specified when configuring the ODBC connection (in my case, I called mine ICED-ALPINE). And then of course the user name and password are the credentials for the ”reader” account that has read privileges only.

After you hit enter, Matlab should take a minute to connect to the database, but when it does, it should spit out some code in the command window essentially saying you are connected. You will know if you are not connected if, under the message: section it gives you some sort of error.

Anyway, once you are connected, you should be able to query, extract, display and analyze data from the database directly in Matlab! Nice. If you are struggling to connect, please feel free to reach out either to Greg (balcs@bgc.org) or Joe (jtulenko@bgc.org).

Thanks for following along!