

Guide for connecting MySQL database to IntelliJ

1. Install the community version of IntelliJ at this link:

<https://www.jetbrains.com/idea/download/#section=windows>

Download IntelliJ IDEA

Windows

macOS

Linux

Ultimate

For web and enterprise development

Download

.exe



Free 30-day trial available

Community

For JVM and Android development

Download

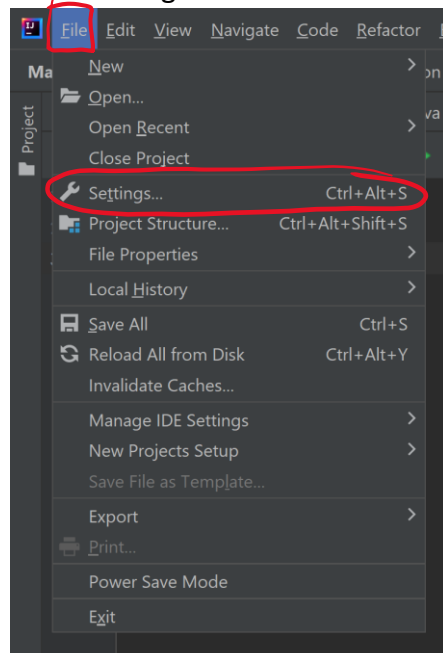
.exe



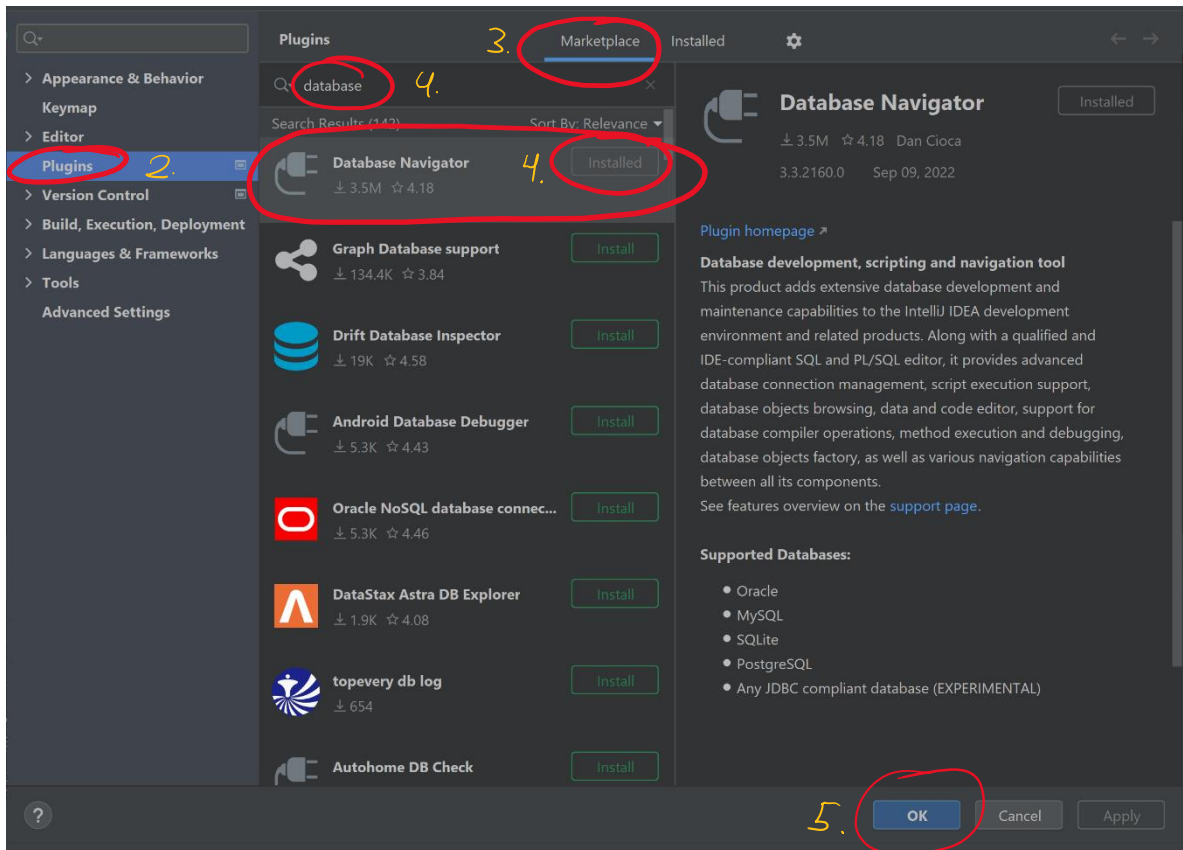
Free, built on open source

Installing the Database Navigator Plugin

1. Go to File >> Settings...

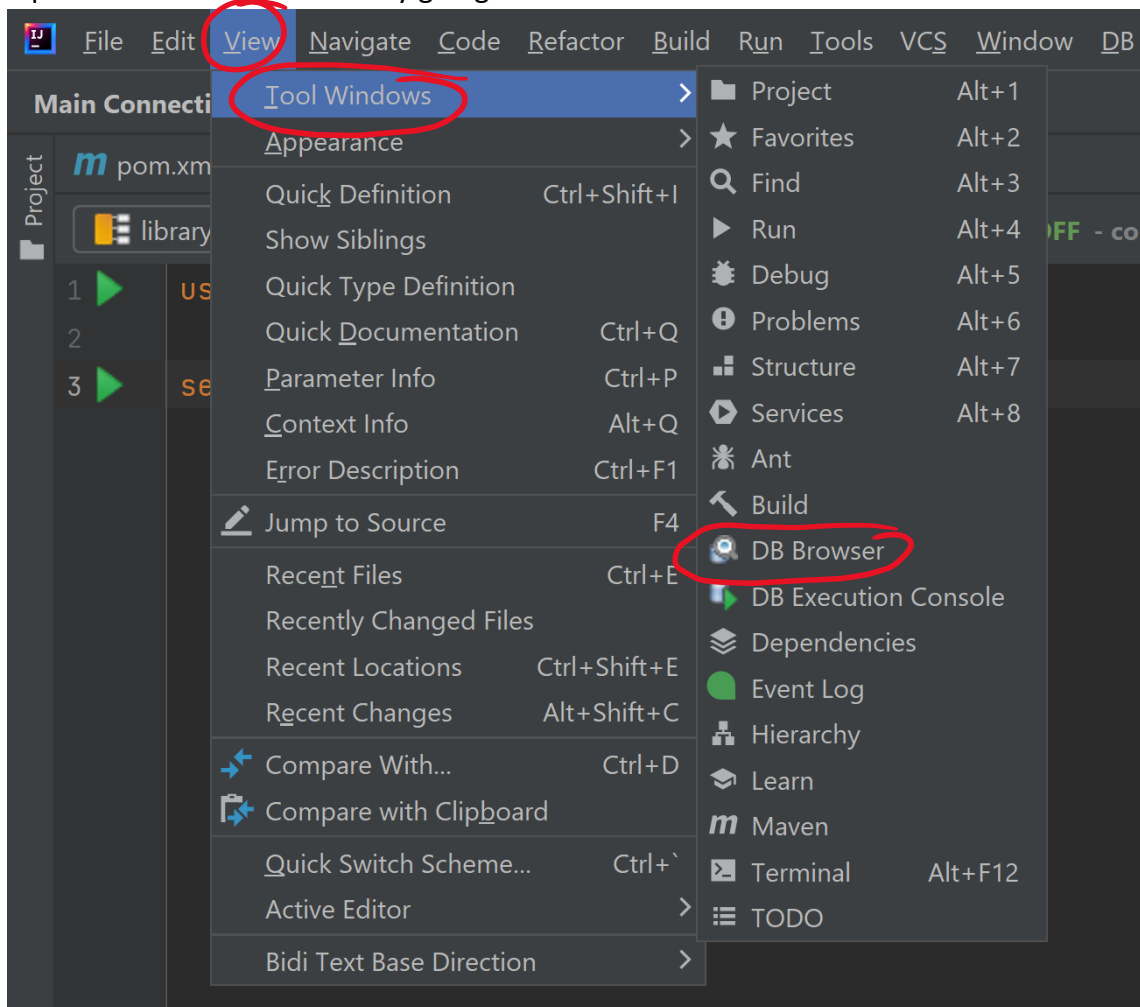


2. A window will pop up. Click on the Plugins button on the left.
3. On the top of the Settings window, select the Marketplace tab.
4. Search for “database navigator” and install the first one on the list.
5. Then click “ok” on the bottom. (The IDE will need to be restarted after installing this plugin. It will give you a button to do so if after clicking “Install”.)

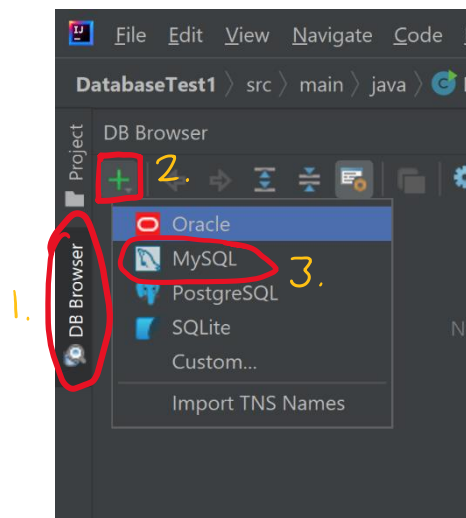


Creating a connection with Database Navigator

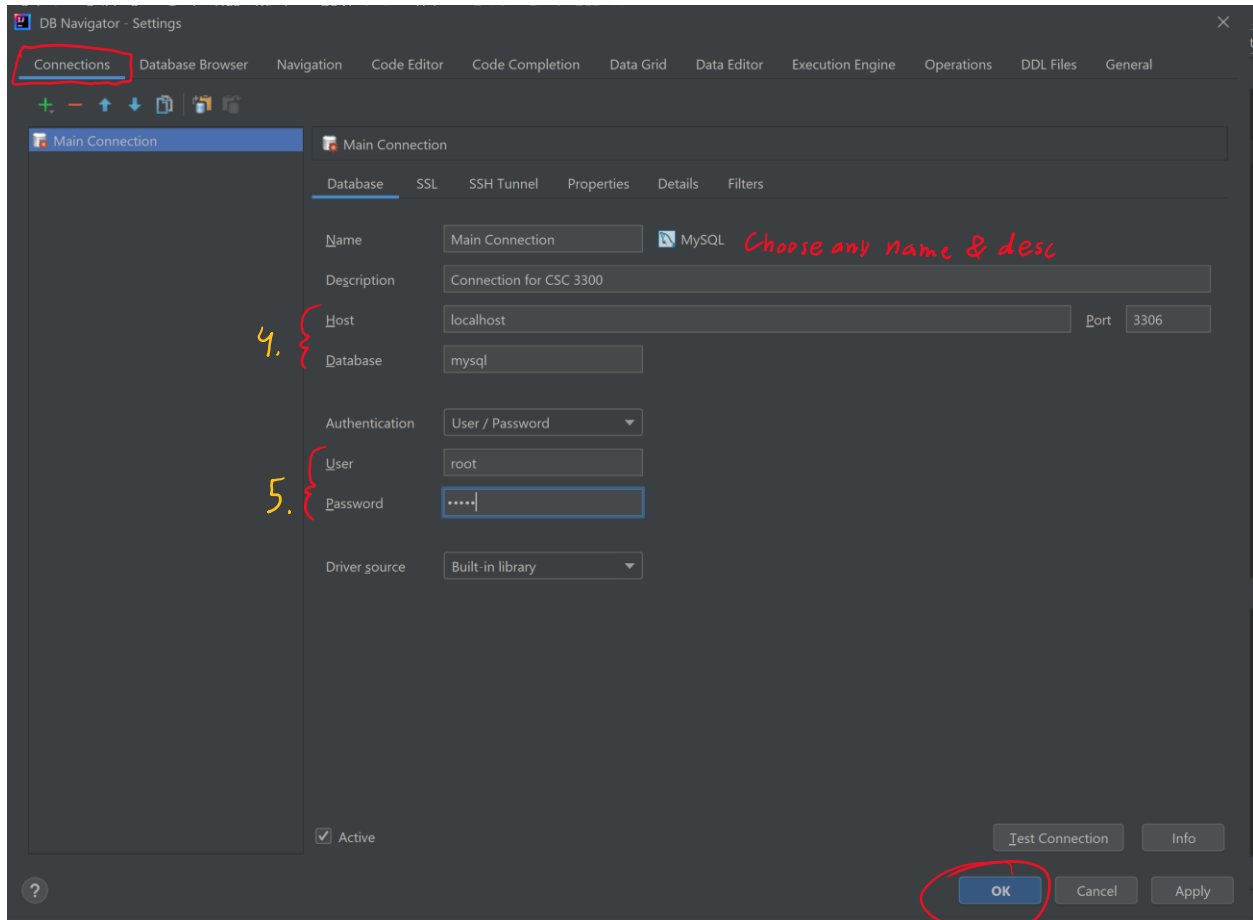
1. Open the database browser by going to View >> Tool Windows >> DB Browser



2. In the DB Browser tab, click on the green + icon on the top left, and select "MySQL"

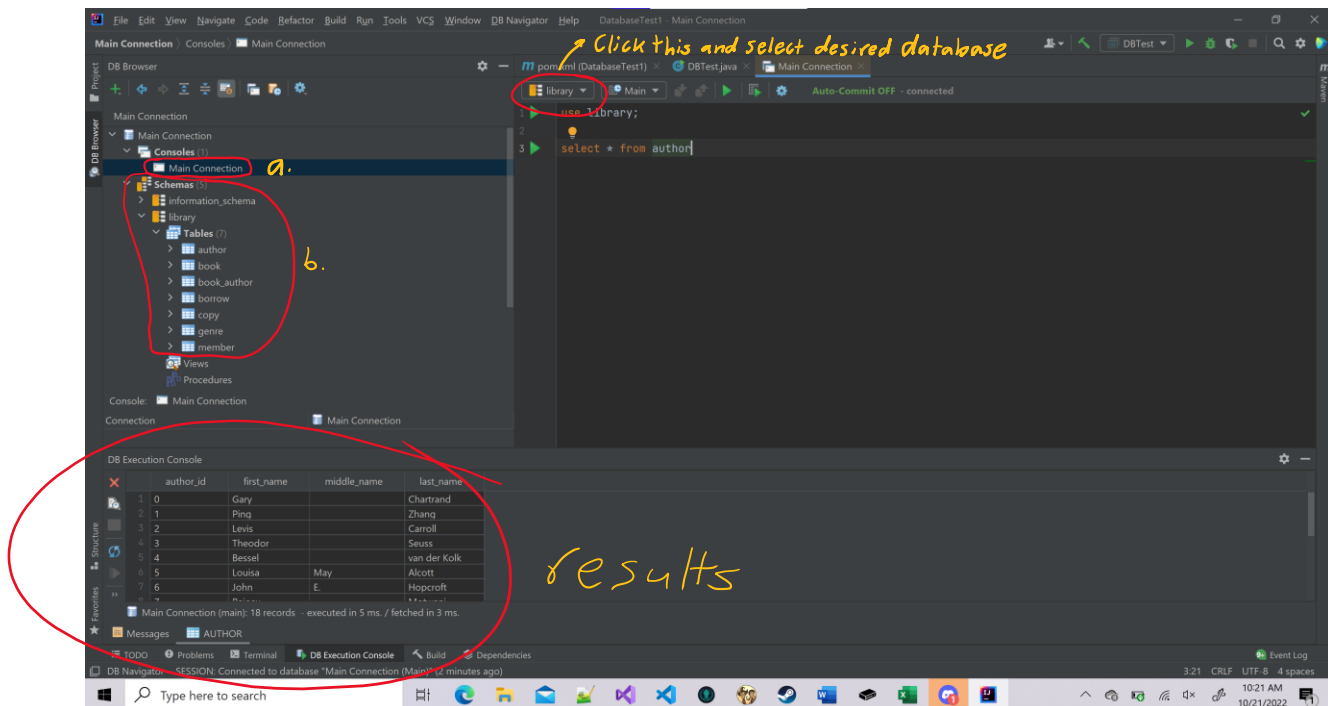


3. Before this next step, make sure your database (mysqld.exe) is running for this step to work!
4. In the connections window that pops up, you can name the connection anything you'd like. Since we're using databases only on our local computers, you can leave the Host: localhost and Database boxes alone.
5. For the username and password, unless you changed the username or created a different password during the mysql setup, enter "root" as the username and "db123" as the password (unless you chose a different password). Then hit "ok".



Interacting with the database connection

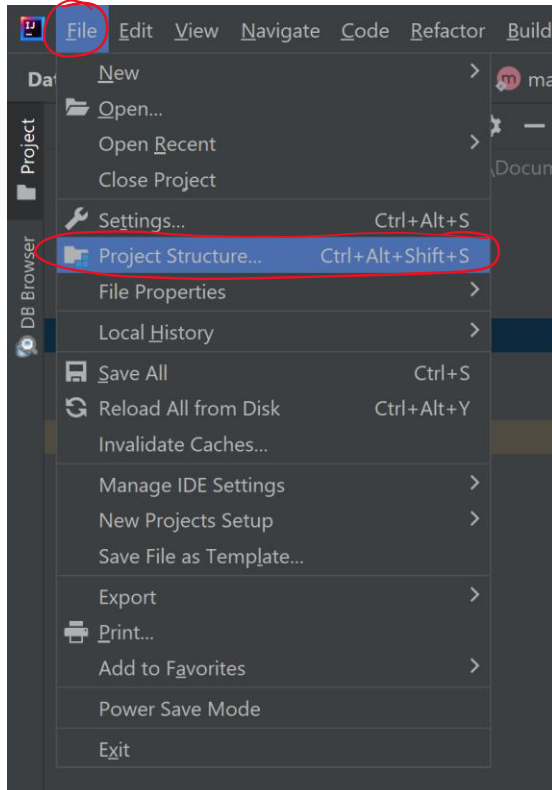
- Double click on whatever entry is under the “Consoles” tab to open a window that will allow you to write MySQL scripts. (I named mine “Main Connection” earlier, so the console in the screenshot has the same name.)
- On the left under Schemas, you should be able to see the different “databases” you’ve already created (like library or university) along with some other hidden stuff we don’t use (like information_schema).
- In order to execute an SQL script with this plugin in IntelliJ, you must first select what database you are using on the top. Results should appear on the bottom.



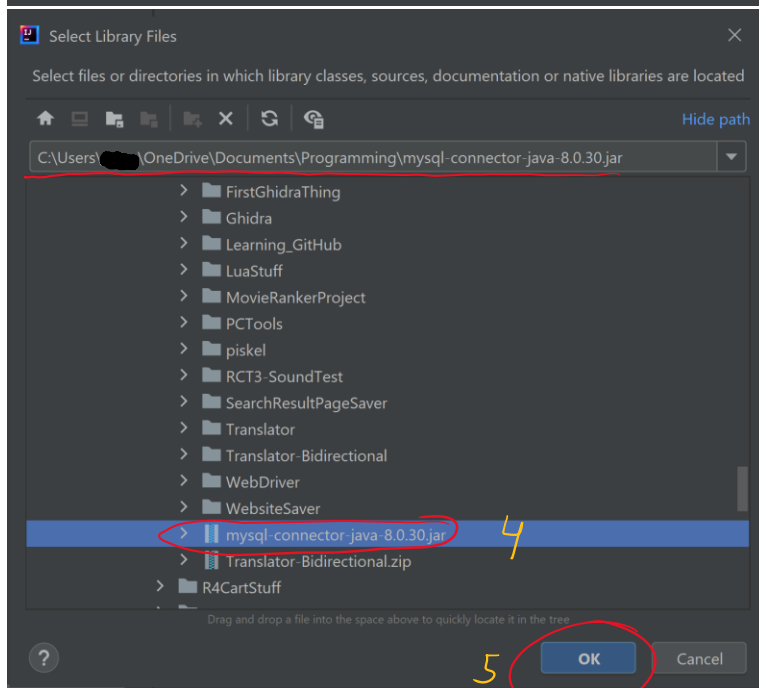
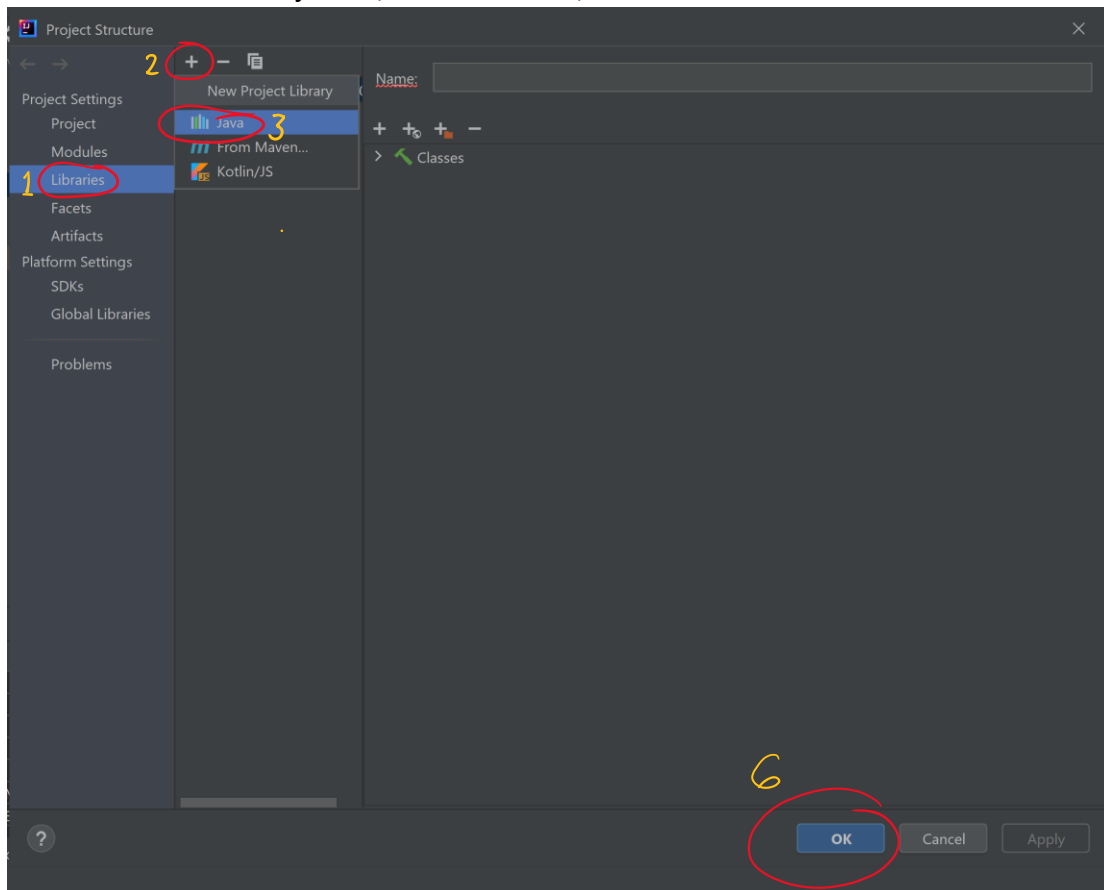
This guide basically explains how to use IntelliJ in place of MySQL Workbench, I can make a separate guide on how to communicate with a database with Java itself later if the class fails to do that clearly...

Using Java to communicate with the database

1. Download the latest version of the mysql connector .jar file here:
 - a. <https://mavenlibs.com/jar/file/mysql/mysql-connector-java>
2. Move this .jar file to the folder of your choosing.
3. In a new blank project in IntelliJ, go to the project structure in File >> Project Structure...



4. In the new window that appears, click on Libraries on the left, and then click the “+” button on the top and select “Java”. In the explorer window, go to the folder where you stored the connector .jar file, select that file, and select “Ok”.



5. In order to make sure that the connection works in Java, I created a small test in the main function of the Java program to query the first name from the author table in the library schema. (Note: The username and password used here on line 10 in the getConnection function should be the same username and password you used earlier to establish the connection to your database in the database navigator.) Since my database is in the default state (with the small database insert file), I got "Gary" as my test result. Now I can be sure that my connection AND query are working.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class DBTest {
    public static void main(String[] args) {
        try
        {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/library","root","db123");
            Statement stmt = conn.createStatement();
            ResultSet res = stmt.executeQuery("select * from author");
            res.next();
            System.out.println(res.getString("first_name"));
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
        System.out.println("yo this works!");
    }
}
```

