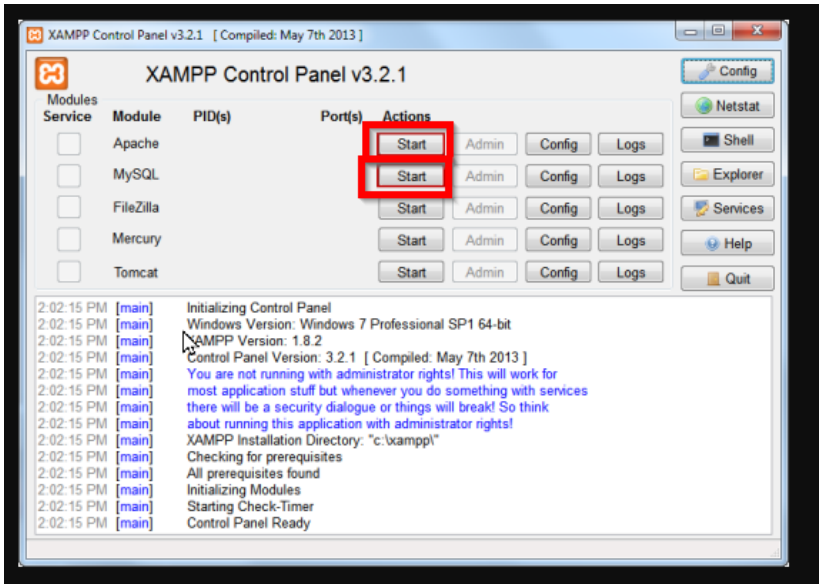
Install and preparing mysql DB

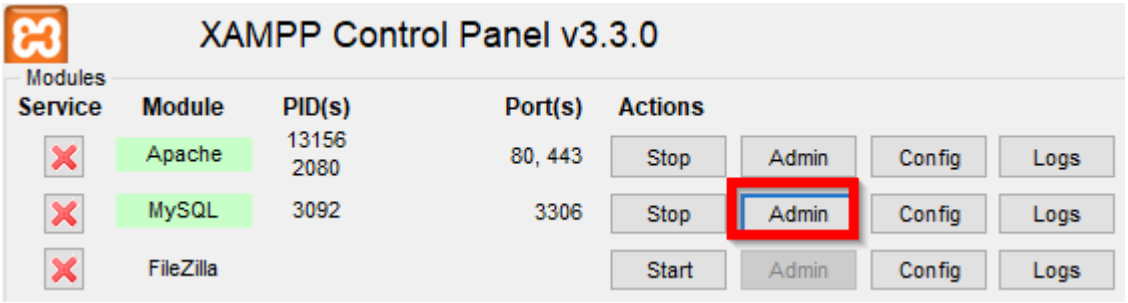
Download and install XAMPP -> https://www.apachefriends.org/download.html

Start XAMPP Control Panel

Start Apache and Mysql



On Mysql select Admin button for phpMyAdmin panel, should open in default browser

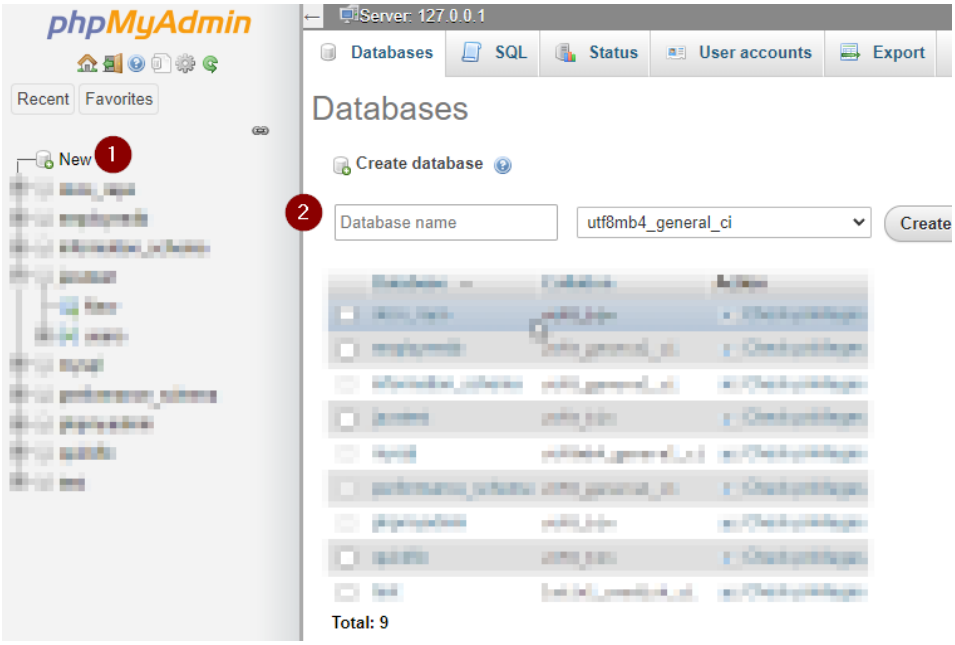


NOTE!!

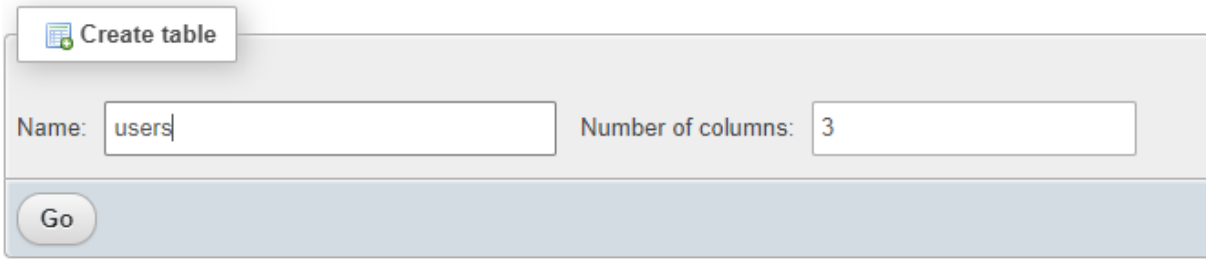
If XAMPP freezes or crashes while stoping Apache and MySql modules, kill process in Task Manager,

next time start Xamp Control Panel with „Run as administrator“ privileges.

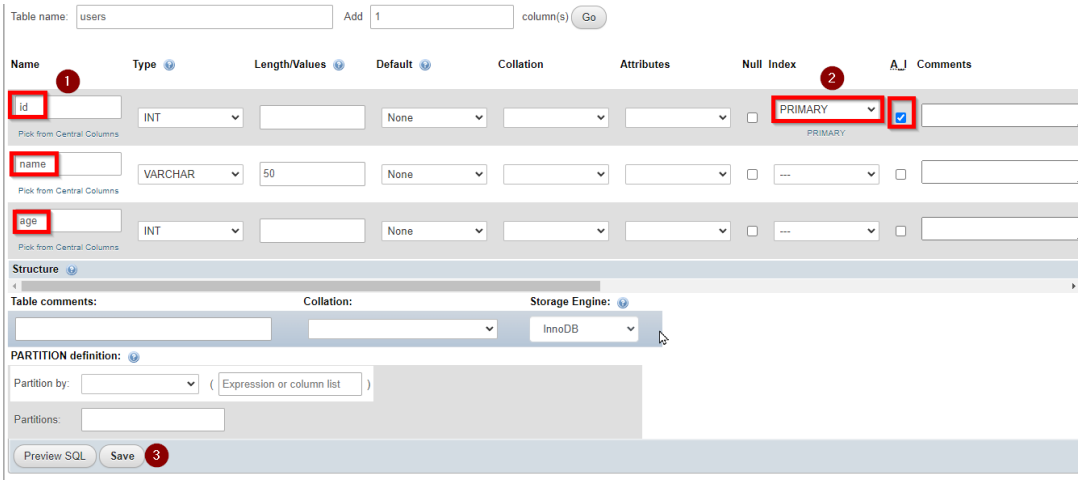
Select New DB, add name (example, „javatest“) and confirm on Create



Add new table name and number of columns and confirm on **Go**



set column properties and confirm on **Save**



Connection settings to DB

Default DB connect properties are:

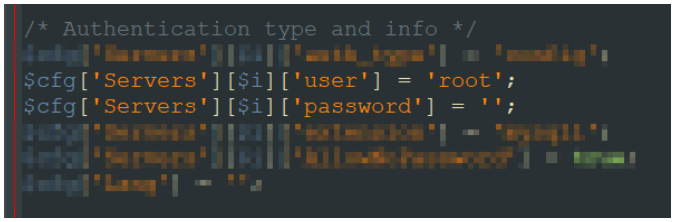
username: root,

password: [empty] (no password set),

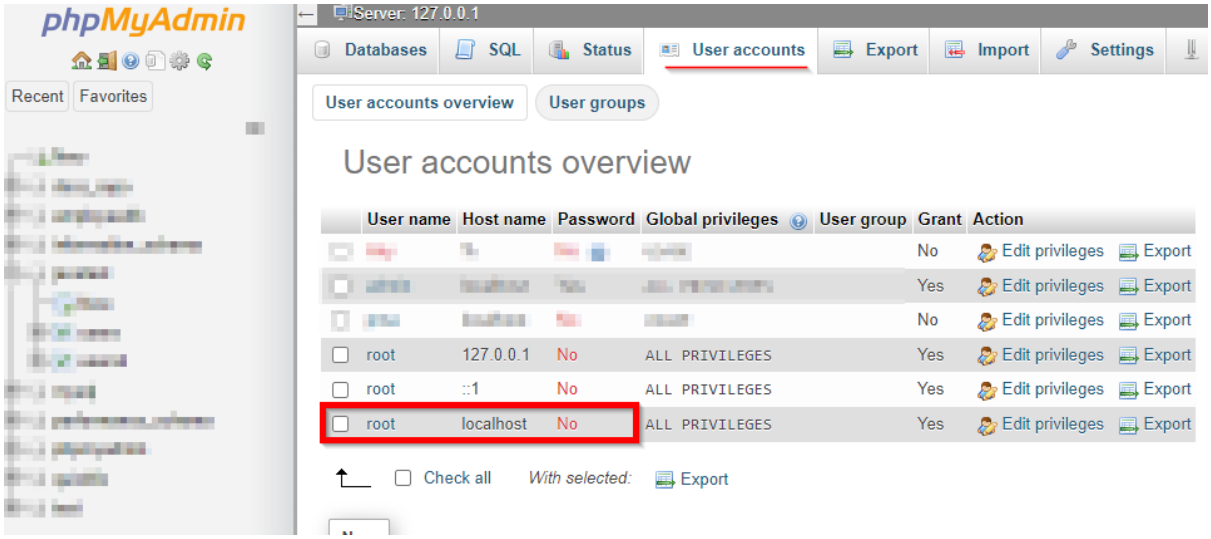
host: localhost,

port: 3306

additionaly can be edited in **config.inc.php** file in **C:\xampp\phpMyAdmin** location



or at **User accounts** of MySql **phpMyAdmin** dashboard,



or add new user

example,

user: **admin**,

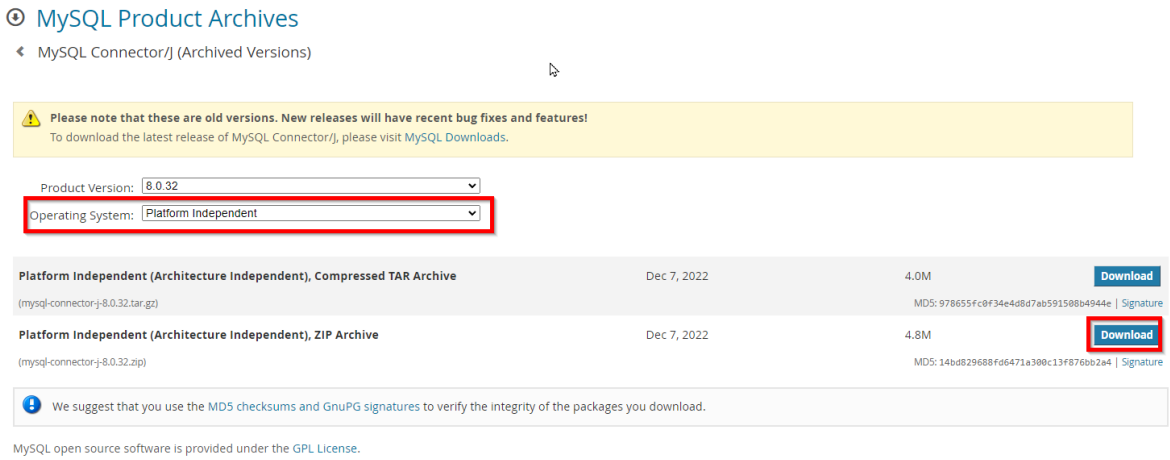
password: **admin**

this credentials are used to establish connection to DB.

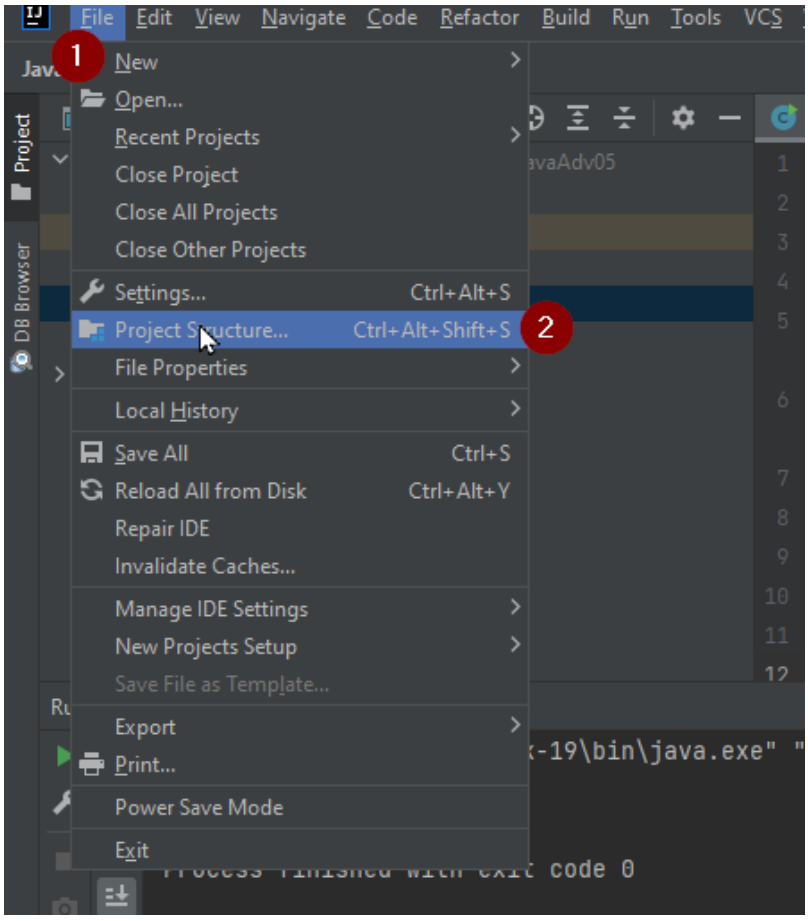
Adding mysql jdbc driver to intelliJ

Download driver for mysql: https://downloads.mysql.com/archives/c-j/ (or https://dev.mysql.com/downloads/connector/j/)

Extract jar file from zip to custom folder



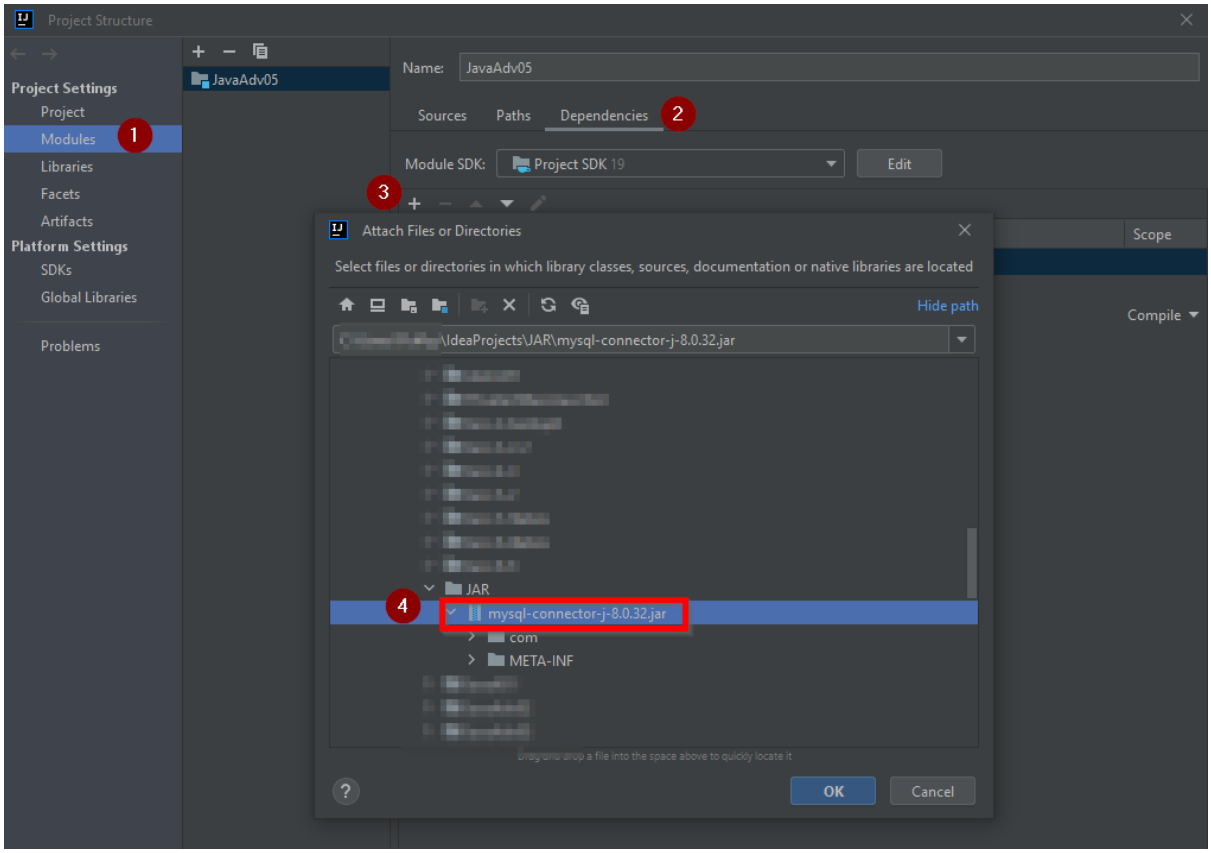
In IntelliJ select File -> Project Structure



In Project structure settings select „Modules“ section -> „Dependencies“ tab

Select **+** (Add) button -> „JARs and Directories“

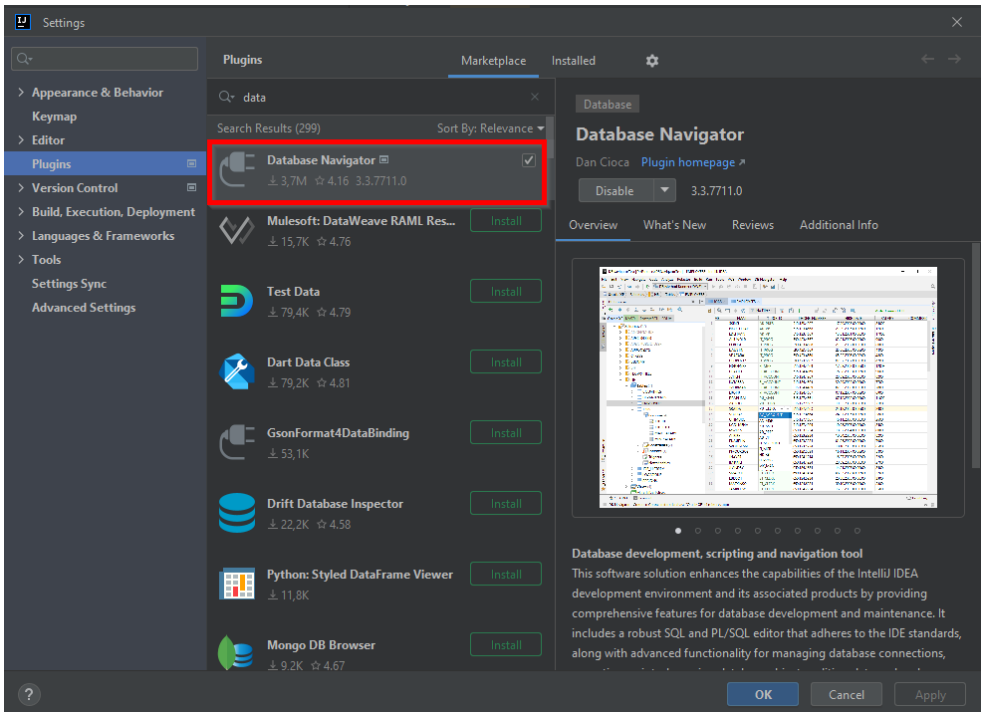
in **Attach Files or Directories** dialog, find location of unzipped file and select **mysql-connector-j-x.x.xx.jar** file, and confirm „Ok“ dialog and project structure dialog window.



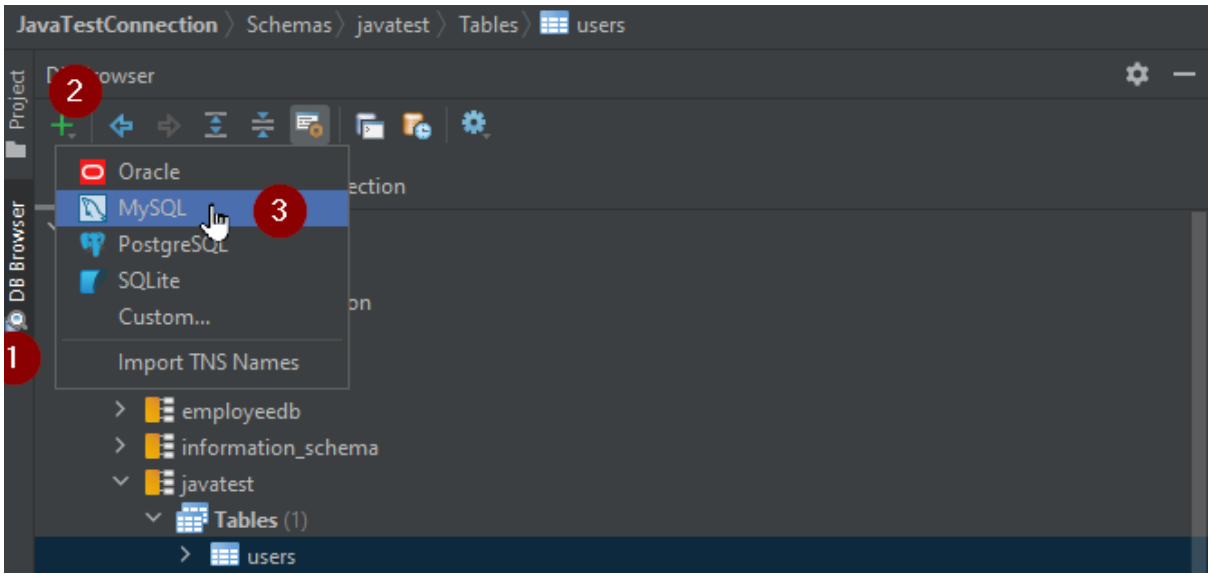
Setting up DB navigator

Open File –> Settings -> Plugins section

On Marketplace search for Database Navigator and Install

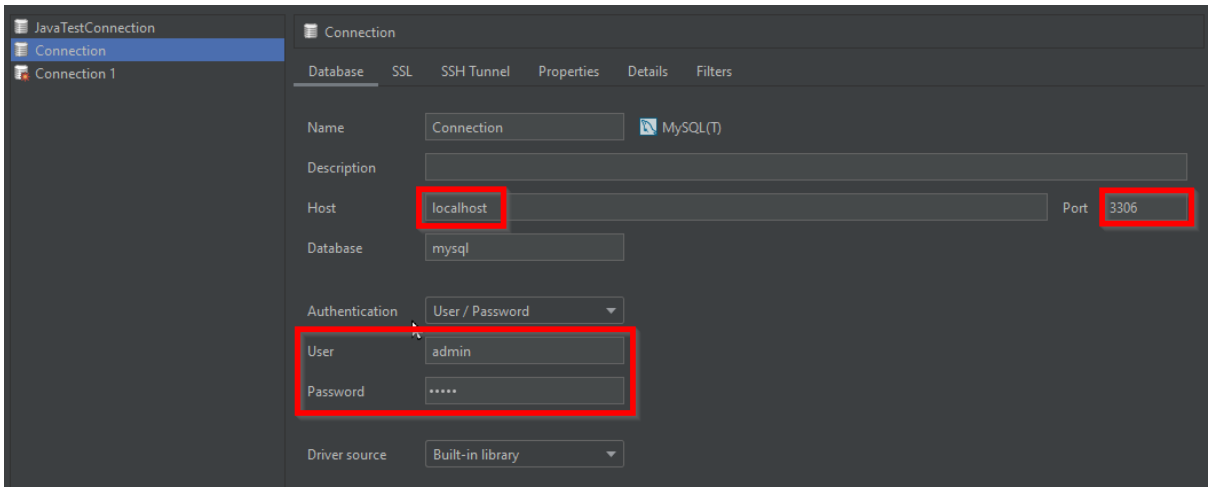


Select DB Browser panel and create new connection

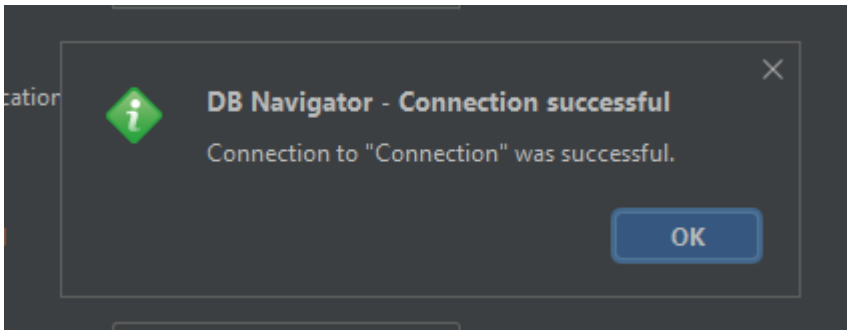


Fill required fields for name, host, port (leave the default port), username and password for connection

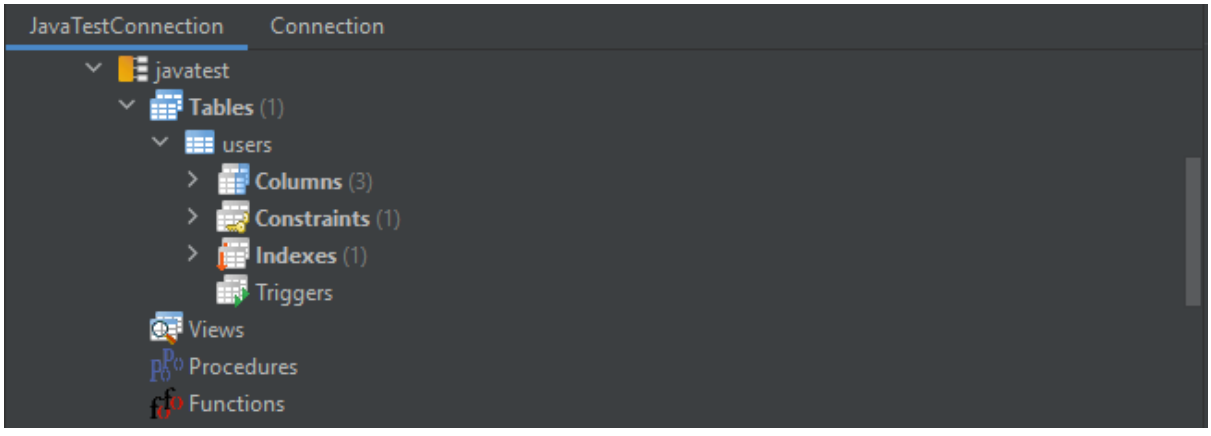
Note: if there is no password set, on „Authentication“ field select: „User“ option



Check on Test connection and Ok to connect



Connection to DB should be established, a should be able to see users table



Inserting first data

Setting properties for connection to DB:

try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/javatest", "admin", "admin")) {

}

```
try (Connection conn = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/javatest",  user: "admin",  password: "admin")) {
```

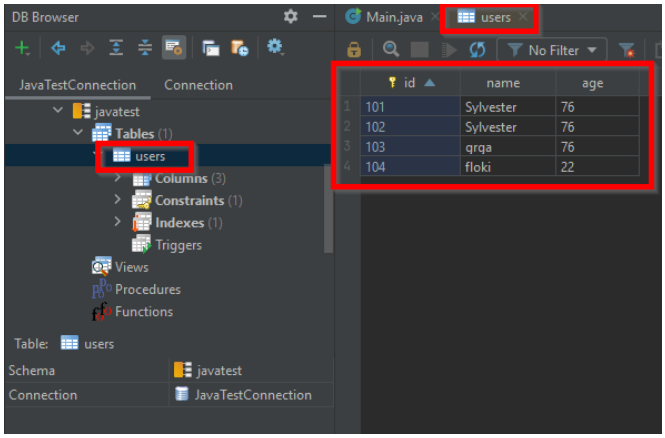
inserting several entries in Main class, one by one

```
public class Main {
    no usages
    public static void main(String[] args) throws SQLException {
        try (Connection conn = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/javatest",  user: "admin",  password: "admin")) {

//            PreparedStatement ps = conn.prepareStatement("INSERT INTO users (id, name, age) VALUES (101,'Sylvester', 76)");
//            PreparedStatement ps = conn.prepareStatement("INSERT INTO users (name, age) VALUES ('Sylvester', 76)");
//            PreparedStatement ps = conn.prepareStatement("INSERT INTO users (name, age) VALUES ('qrqa', 76)");
            PreparedStatement ps = conn.prepareStatement( sql: "INSERT INTO users (name, age) VALUES ('floki', 22)");

            System.out.println("SQL statement successful." + ps.executeUpdate());
        }
    }
}
```

checking for data in DB Browser



checking for data in phpMyAdmin